School of Computing
FACULTY OF ENGINEERING & PHYSICAL SCIENCES

UNIVERSITY OF LEEDS

COMP3911 Secure Computing

5: Public Key Cryptography

Nick Efford

https://comp3911.info

Slides are customised for XJCO3911 by Turgay Celik (Turgay.Celik@swjtu.edu.cn)

Objectives

UNIVERSITY OF LEEDS

- For you to understand the basic principles underlying **public key cryptography** (PKC)
- For you to appreciate the differences between PKC and symmetric ciphers
- For you to recognise the applications of PKC to key distribution and message authentication

Introduction to PKC

UNIVERSITY OF LEEDS

- Different from symmetric ciphers
  - Based on number theory, not bit manipulation
  - Uses *two* keys: a **public key** (+) and a **private key** (−), with a specific mathematical relationship
  - One key is for encryption, the other for decryption – so these are **asymmetric ciphers**
- Common misconceptions:
  - “PKC is more secure than symmetric encryption”
  - “PKC makes symmetric ciphers obsolete”

Encryption Scenario

UNIVERSITY OF LEEDS

Eve

K_{B+} doesn't help with decrypting C

Alice

Bob

C

Has P and K_{B+}

$C \longleftarrow E_{B+}(P)$

Has C and K_{B-}

$P \longleftarrow D_{B-}(C)$

Mallory

K_{B+} doesn't help with decrypting C but msg from Alice could be faked

Requirements

UNIVERSITY OF LEEDS

- Easy to encrypt, given K_+
- Easy to decrypt, given K_-
- Infeasible to determine K_-, given K_+
- Infeasible to recover P, given K_+ and C

Classic Approach: RSA

UNIVERSITY OF LEEDS

- Named after its inventors (Rivest, Shamir, Adleman)
- Well-established (published in 1977)
- Widely used in industry
- Security based on the extreme difficulty of factoring the product of two very large prime numbers

Elliptic Curve Cryptography

UNIVERSITY OF LEEDS

- Newer, increasingly popular approach, based on the mathematical properties of elliptic curves
- Equivalent security to RSA, for smaller key sizes
  - Example: 3072-bit RSA ~ 256-bit ECC
- Creating keys and generating digital signatures is faster than RSA (but verifying signatures is slow)
- Theoretically much more vulnerable to attacks using quantum computing (not practical yet)

(see this Cloudflare blog for an easy-to-follow intro to ECC)

But there is controversy over an NSA **backdoor** in an element of ECC…

The New York Times 13 Sept 2013

But internal memos leaked by a former N.S.A. contractor, Edward Snowden, suggest that the N.S.A. generated one of the random number generators used in a 2006 N.I.S.T. standard — called the Dual EC DRBG standard — which contains a back door for the N.S.A. In publishing the standard, N.I.S.T. acknowledged "contributions" from N.S.A., but not primary authorship.

Internal N.S.A. memos describe how the agency subsequently worked behind the scenes to push the same standard on the International Organization for Standardization. "The road to developing this standard was smooth once the journey began," one memo noted. "However, beginning the journey was a challenge in finesse."

NIST SP 800-90

Acknowledgements

The National Institute of Standards and Technology (NIST) gratefully acknowledges and appreciates contributions by Mike Boyle, Paul Timmel and Debby Wallner from the National Security Agency for assistance in the development of this Recommendation. NIST also thanks the many contributions by the public and private sectors, and by the Cryptographic Tool Standards and Guidelines working group of American Standards Committee X9, whose thoughtful and constructive comments improved the quality and usefulness of this publication.

RSA Algorithm

UNIVERSITY OF LEEDS

1. Select two large prime numbers, p and q
2. Calculate the product $n \longleftarrow pq$
3. Calculate **Euler totient** of n, $\varphi(n) \longleftarrow (p-1)(q-1)$
4. Select an integer e, such that greatest common divisor of e and $\varphi(n)$ is 1
5. Calculate integer d such that $de \bmod \varphi(n) = 1$

$K_+ = \{e, n\}$
$K_- = \{d, n\}$

$C \longleftarrow P^e \bmod n$
$P \longleftarrow C^d \bmod n$

RSA Example

UNIVERSITY OF LEEDS

1. Select two prime numbers

   $p = 17, q = 11$

1. Calculate product

   $n = 17 \times 11 = 187$

1. Calculate Euler totient

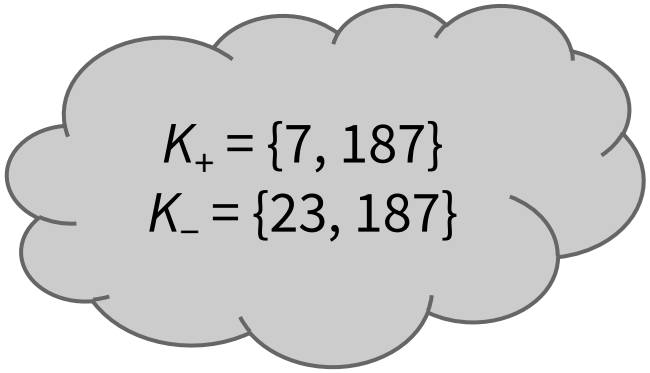
   $\varphi(n) = (p-1)(q-1) = 16 \times 10 = 160$

1. Choose integer e

   $e = 7$

1. Determine value of d

   $7d \bmod 160 = 1 \Rightarrow 7d = 161 \Rightarrow d = 23$



Attacking RSA

UNIVERSITY OF LEEDS

- Brute-force search of all private keys
  - Defeated if *d* and *n* are large enough, but this is what makes encryption & decryption slow!
- Find prime factors of *n*
  - Explored by RSA Security via ‘Factoring Challenges’
  - Challenges used **RSA numbers**: a list of semiprimes of varying sizes, from 100 to 617 decimal digits
  - Challenges stopped in 2007 but some researchers continue with factorisation attempts

RSA Number Examples

UNIVERSITY OF LEEDS

Number	Digits	Bits	Factored?
RSA-100	100	330	April 1991
RSA-129	129	426	April 1994
RSA-576	174	576	December 2003
RSA-768	232	768	December 2009
RSA-240	240	795	November 2019
RSA-250	**250**	**829**	**February 2020**
RSA-1024	309	1024	No
RSA-2048	617	2048	No

RSA-250

UNIVERSITY OF LEEDS

```
21403246502407449612644230728393335630086147151447550177977549208814180234471401366433455190958046796109928518724709145876873962619215573630474547705208051190564931066876915900197594056934574522305893259766974716817380693648946998715784949759374979377
```

==

```
64135289477071580278790190170577389084825014742943447208116859632024532344630238623598752668347708737661925585694639798853367
```

×

```
33372027594978156556226010605355114227940760344767554666784520987023841729210037080257448673296881877565718986258036932062711
```

Computational cost: ~2,700 core-years

Million Messages Attack

UNIVERSITY OF LEEDS

- Discovered by Daniel Bleichenbacher in 1998
- Relies on **malleability** of RSA: certain modifications to ciphertext will successfully decrypt
- If we can observe whether decryption of modified ciphertext was successful, this helps to narrow down the range of possible values for the plaintext
- Sending several million messages is enough to reveal the exact value of the plaintext!
- Fix: OAEP (‘Optimal Asymmetric Encryption Padding’)

CVE-2017-15361

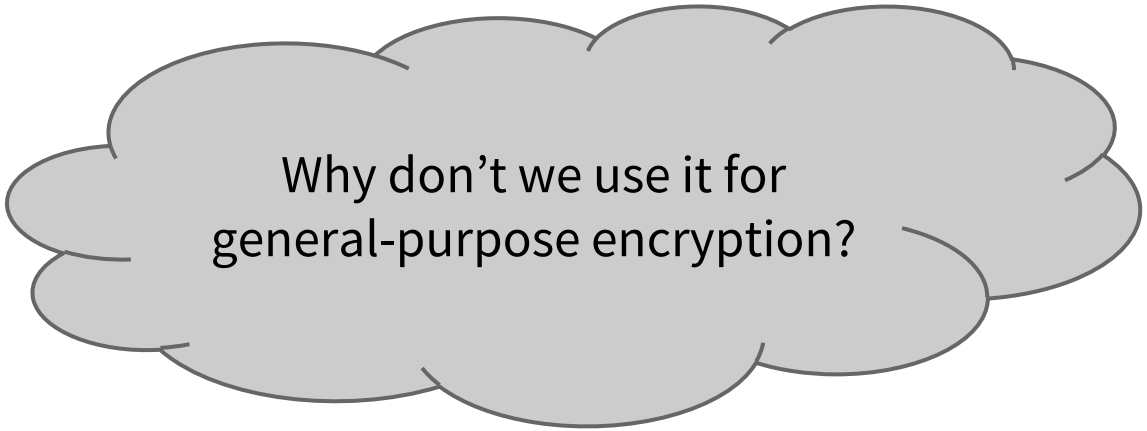
UNIVERSITY OF LEEDS

- Vulnerability in RSA key pair generation
- Found in cryptographic chips used in a range of hardware (smartcards, authentication tokens, OS Trusted Boot systems, etc)
- At least 760,000 keys known to be affected
- Impact was to reduce time required for factorization:
  - 2 core-hours for 512-bit keys
  - 97 core-days for 1024-bit keys
  - 140.8 core-years for 2048-bit keys

Applications of PKC

UNIVERSITY OF LEEDS

- **Hybrid encryption** schemes
  - Example: TLS (discussed in a later lecture)
- Digital signatures



Hybrid Encryption

UNIVERSITY OF LEEDS

- Do AEAD with a symmetric cipher and a MAC
  - e.g., using AES-GCM
- Use a public key cryptosystem to protect the keys needed by the cipher and the MAC
- Gives the high performance of a symmetric cipher while solving the problem of exchanging keys securely
- Practical example (in Java): Exercise 7

Digital Signatures

UNIVERSITY OF LEEDS

- What if Alice encrypts something using K_{A-}?
- No good for confidentiality, because anyone can decrypt using K_{A+} …
- … but the decrypted data can only have come from Alice (**assuming K_{A-} is secure!**)
- So if Alice encrypts the hash of a message using K_{A-}, authenticity and integrity are assured
- Private key-encrypted hash can act a **digital signature**

Ed25519

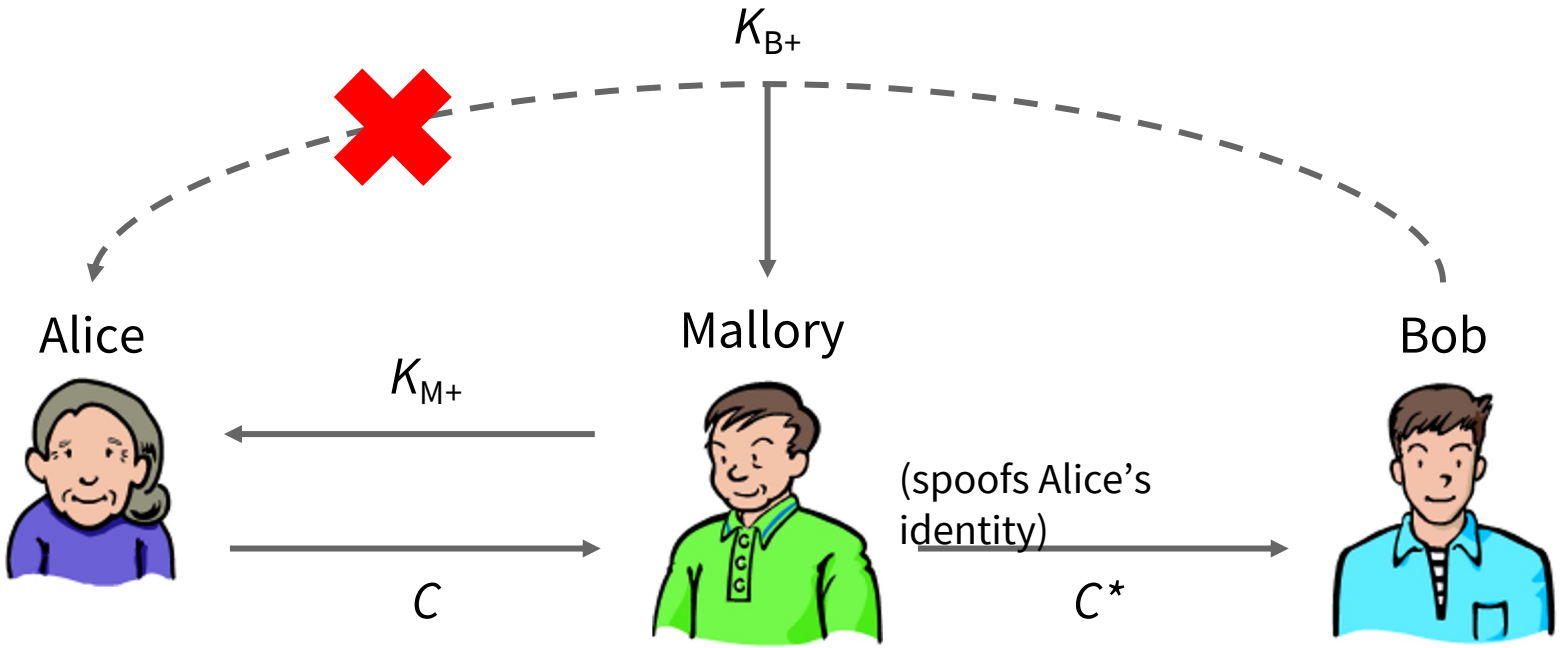
UNIVERSITY OF LEEDS

- Instance of the Edwards curve digital signature algorithm, using SHA-512 and a specific elliptic curve
- **Very fast at signing**
- Small public keys (32 bytes compressed) and small signatures (64 bytes compressed)
- Resistant to hash function collisions
- Immune to **side-channel attacks** involving CPU cache
- Supported authentication option in SSH

Public Key Trust

UNIVERSITY OF LEEDS

- Suppose Alice receives an email from Bob, containing his public key, K_{B+}
- Alice can use this to share a symmetric key securely with Bob, or verify a digital signature Bob has made
- BUT can she trust that K_{B+} genuinely came from Bob?
- Solutions:
  - Face-to-face verification
  - Public key certificates



Has K_{M+} thinking it is K_{B+} so computes

$C \longleftarrow E_{M+}(P)$

Has K_{M-} so computes

$P \longleftarrow D_{M-}(C)$

Has K_{B+} so computes

$C^* \longleftarrow E_{B+}(P)$

Public Key Certificates

UNIVERSITY OF LEEDS

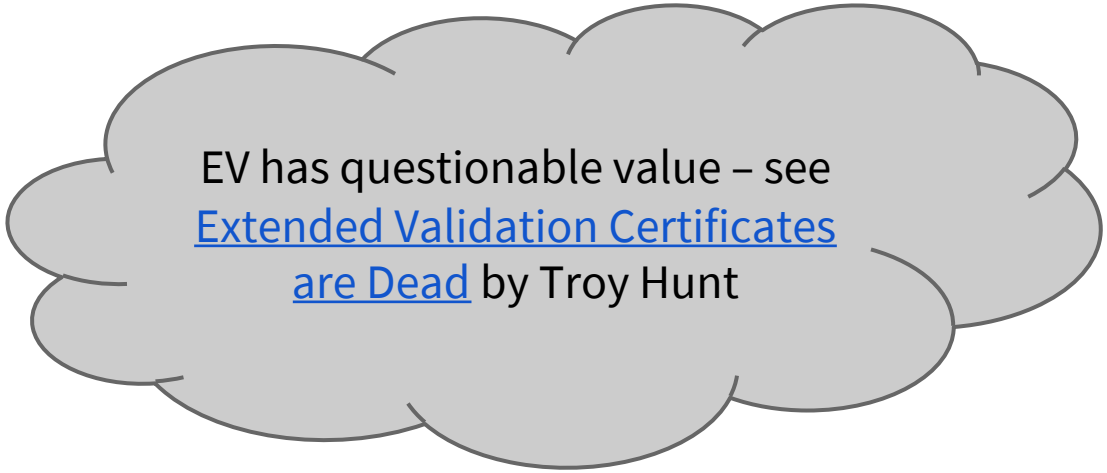
- Public key and owner ID can be transported in a **public key certificate**, signed by a **certificate authority**
- **CA is a trusted third party**, in theory taking responsibility for performing checks on the key's owner
- Example: **TLS certificates** for HTTPS
  - Server provides browser with its public key in a certificate, signed by a CA
  - Browser uses CA's public key to verify certificate

TLS Certificate Validation

UNIVERSITY OF LEEDS

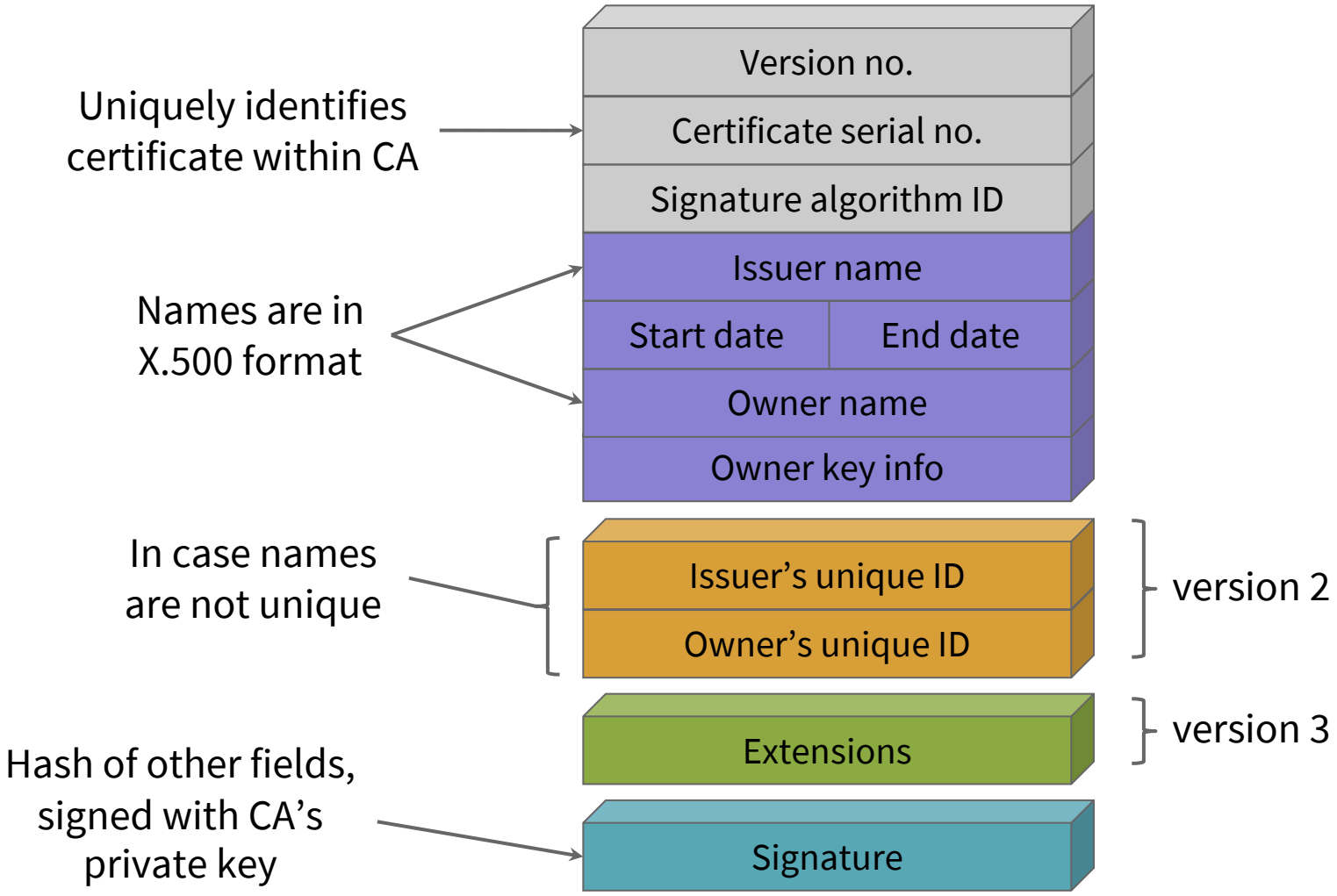
Three levels:

- DV (domain validation)
- OV (organisation validation)
- EV (extended validation)



X.509 Certificate Format

UNIVERSITY OF LEEDS



Let’s Encrypt

UNIVERSITY OF LEEDS

- https://letsencrypt.org
- “A free, automated, open certificate authority”
- Makes it possible to set up an HTTPS server and have it automatically obtain a browser-trusted certificate, without any human intervention
- Automation a key benefit, since it prevents the alarmingly regular problem of forgetting to renew a certificate!
- Issued certs are valid for only 90 days, limiting the damage caused by mistakes, stolen keys, etc

Summary

UNIVERSITY OF LEEDS

We have

- Seen that public key cryptography involves use of a public key for encryption & private key for decryption
- … or a private key for creating a digital signature and a public key for signature verification
- Noted PKC's inefficiency and seen how its role is limited to encrypting/signing small things (keys & hashes)
- Explored details of the RSA algorithm and considered its resistance to attack
- Discussed the issue of public key trust and how this is achieved through the use of certificates

Follow-Up / Further Reading

UNIVERSITY OF LEEDS

- Code Examples
- Exercise 7, Exercise 8 and Exercise 9
- RSA-768 paper: “Factoring of a 768-bit modulus” http://eprint.iacr.org/2010/006.pdf
- Article on factoring of RSA-240
- ROCA: Vulnerable RSA Generation (CVE-2017-15361)
- “Breaking the encryption scheme of the Moscow internet voting system” (August 2019) https://arxiv.org/abs/1908.05127