Public Key

Basics
RSA/ECC
Applications (Encryption and Signing)

Prof Bill Buchanan OBE

https://asecuritysite.com/rsa
https://asecuritysite.com/ecc
https://asecuritysite.com/elgamal

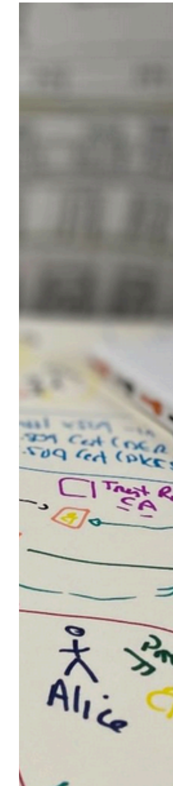
Publi

Basics
RSA/ECC
Applicatio

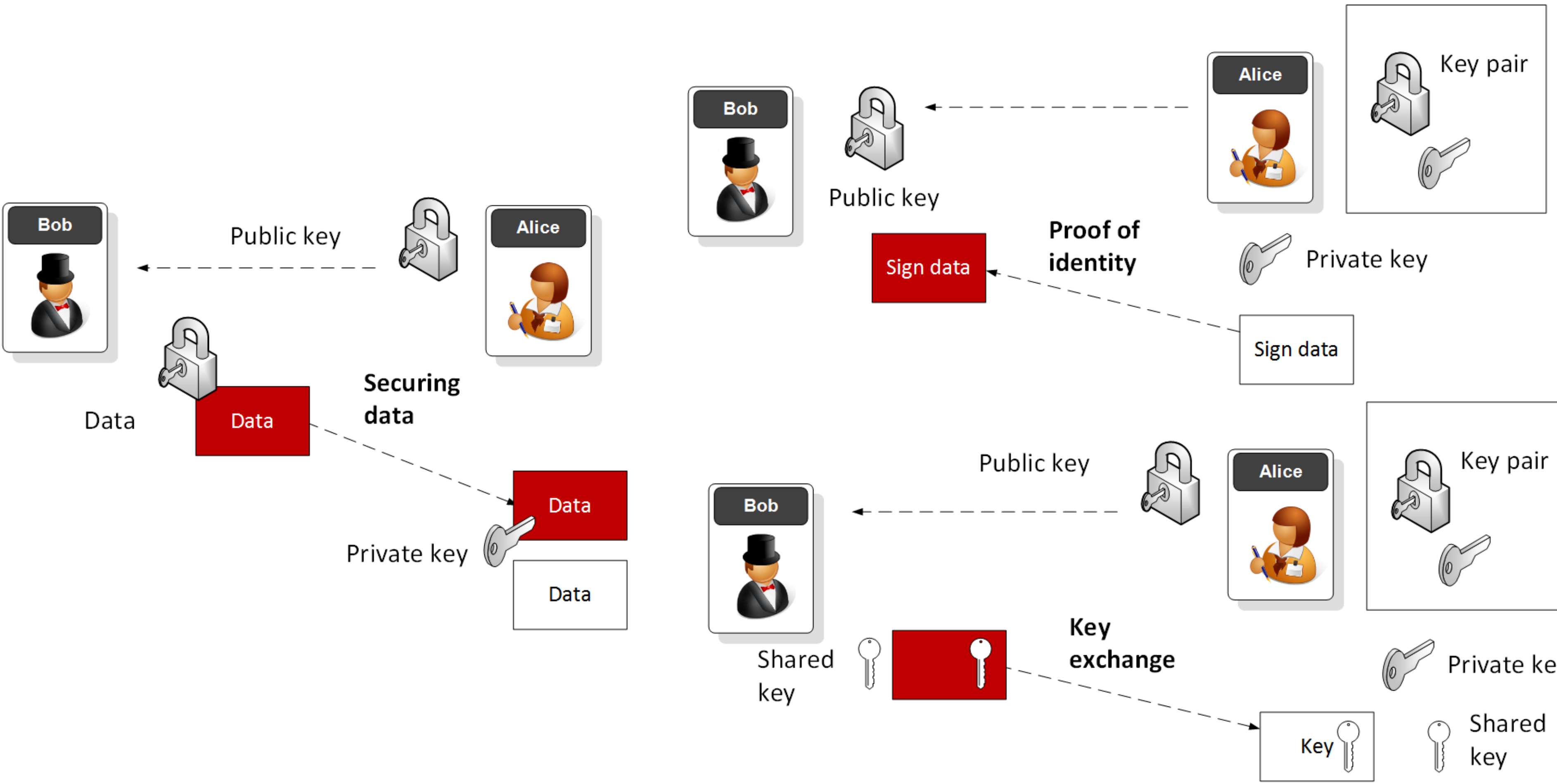
Prof Bi

https://ase
https://ase
https://ase

No	Date	Subject	Lab
2	13 Sept 2023	1. Introduction [Link] 2. Intrusion Detection Systems [Link]	Introduction to Vyatta Lab
3	20 Sept 2023	3. Network Security [Link]	Vyatta and Snort. [Link]
4	27 Sept 2023	4. Ciphers and Fundamentals [Link]	pfSense.
5	4 Oct 2023	5. Secret Key 6. Hashing [Link]	AWS Security and Server Infrastructures
6	11 Oct 2023	7. Public Key [Link] 8. Key Exchange [Link]	Public/Private Key and Hashing
7	18 Oct 2023	Reading week	Reading week
8	25 Oct 2023	9. Digital Certificates	Certificates here
9	1 Nov 2023	Test 1 here	
10	8 Nov 2023	10 Network Forensics here	Network Forensics lab
11	15 Nov 2023	11. Splunk here	Splunk Lab here
12	22 Nov 2023	13. Tunnelling here	Tunnelling
13	29 Nov 2023	14. Blockchain and Cryptocurrencies here	Blockchain Lab.
14	6 Dec 2023		
15	13 Dec 2023	Hand-in: TBC [Here]	



Public Key Methods

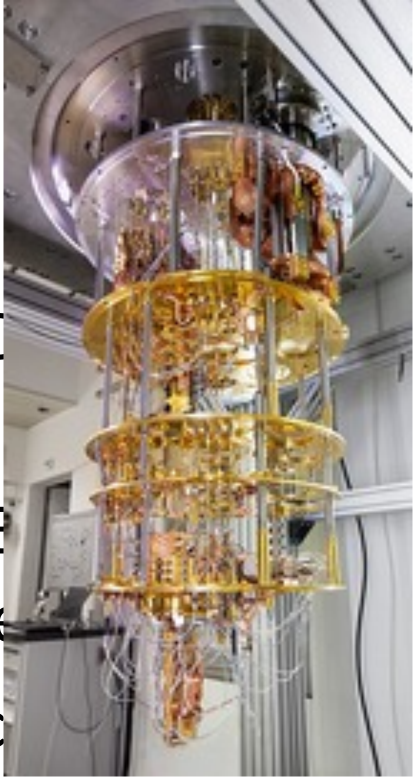


Public Key Methods

- **Integer Factorization**. Using prime numbers. Example: RSA. Key size: 2,048 bits (modulus). Signing, Digital Certificates.
- **Discrete Logarithms**. $Y = g^x \bmod P$. Example: ElGamal. Prime number size: 2,048 bits. Key handshake.
- **Elliptic Curve Relationships**. Example: Elliptic Curve. Private key: 256 bits. Public key: 512 bits. Bitcoin, IoT, Web, etc.

Public Key Methods

- **Integer Factorization**. Using prime numbers. RSA. Key size: 2,048 bits (modulus). Signing, D Certificates.
- **Discrete Logarithms**. $Y = g^x \bmod P$. Example: E Prime number size: 2,048 bits. Key handshake
- **Elliptic Curve Relationships**. Example: Elliptic Private key: 256 bits. Public key: 512 bits. Bitcoin, IoT, Web, etc.



Public Key

RSA

Prof Bill Buchanan OBE

https://asecuritysite.com/rsa
https://asecuritysite.com/ecc
https://asecuritysite.com/elgamal

Eve

p

9,137,187,070,061,098,912,312,979,400,361
,251,189,847,923,809,497,258,114,688,790,
849,334,008,324,856,676,348,809,151,285,1
18,821,829,375,998,699,013,311,467,364,66
2,378,853,216,263,996,490,005,611,058,805

p

9,885,919,140,818,765,444,174,626,190,703
,294,219,553,850,295,249,705,938,896,539,
634,343,302,401,155,295,752,383,276,739,5
84,190,165,200,823,122,225,274,427,125,93
4,163,475,191,779,288,529,189,149,818,011

(p-1)*(q-1)

90,329,492,549,158,751,736,593,291,654,313,033,317,391,509,546,977,632,
830,551,342,194,781,230,803,832,847,247,315,213,556,011,813,523,182,777
,529,551,800,128,685,586,665,697,818,108,995,125,892,738,489,085,065,56
4,398,419,119,705,178,003,889,155,415,914,402,310,708,147,858,313,669,1
76,692,847,865,236,706,085,105,432,191,429,510,583,595,108,030,256,069,
207,938,161,732,170,083,525,341,774,967,620,008,260,040

With Diffie-Hellman we need the other side to be active before we send data. Can we generate a special one-way function which allows is to distribute an encryption key, while we have the decryption key?

Solved in 1977, By Ron Rivest, Adi Shamir, and Len Aldeman created the RSA algorithm for public-key encryption.

Public Key Methods

Public Key

RSA

Pick p and q (two large primes)

$N = p.q$

$PHI = (p-1)(q-1)$

Pick e (no factors with PHI)

$d = InvMod(e,PHI)$

Public: (e,N)
Private: (d,N)

$C = M^e \pmod N$
$P = C^d \pmod N$

ElGamal

Pick random value x, generator (g) and prime (p)

$Y = g^x \pmod p$

Public key: (Y,p)
Private: (x,p)

ECC

Pick random value sk and curve (a, b, G, p, o). G is the base point on curve, p is a prime number, and o is the order of the curve).

For example: $y^2=x^3 + ax + b \pmod p$

$Pk = sk.G$

Public key: Pk
Private key: sk

RSA

- Two primes p, q.
- Calculate N (modulus) as p x q eg 3 and 11. n=33.
- Calculate PHI as (p-1)x(q-1). PHI=20
- Select e for no common factor with PHI. e=3.
- Encryption key [e,n] or [3,33].
- (d x e) mod 20 = 1
- (d x 3) mod 20 = 1
- d= 7
- Decryption key [d,n] or [7,33] (link)

RSA

Calc

Example

- Encryption key [e,n] or [3,33].
- Decryption key [d,n] or [7,33]
- Cipher = M^e mod N

eg M=5.

- Cipher = 5^3 mod 33 = 26
- Decipher = C^d mod N
- Decipher = $(26)^7$ mod 33 = 5

Public Key

Basics

RSA

Applications (Encryption and Signing)

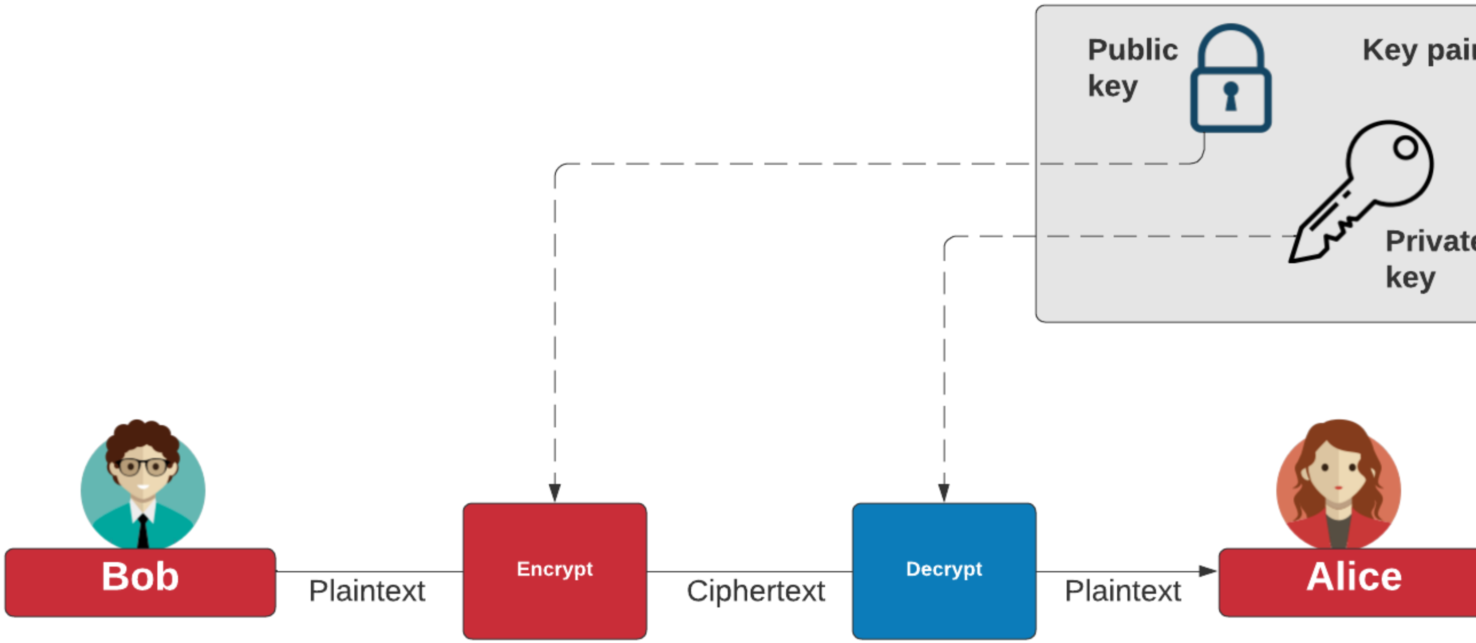
Prof Bill Buchanan OBE

https://asecuritysite.com/rsa

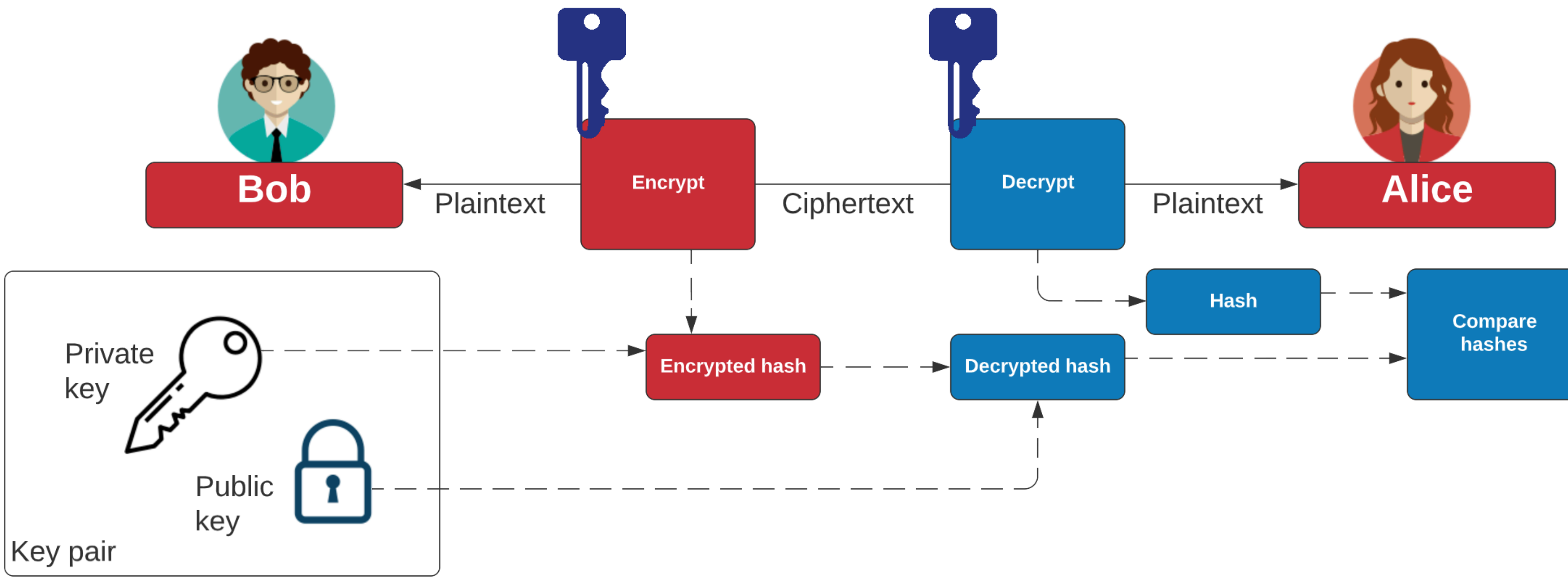
https://asecuritysite.com/ecc

https://asecuritysite.com/elgamal

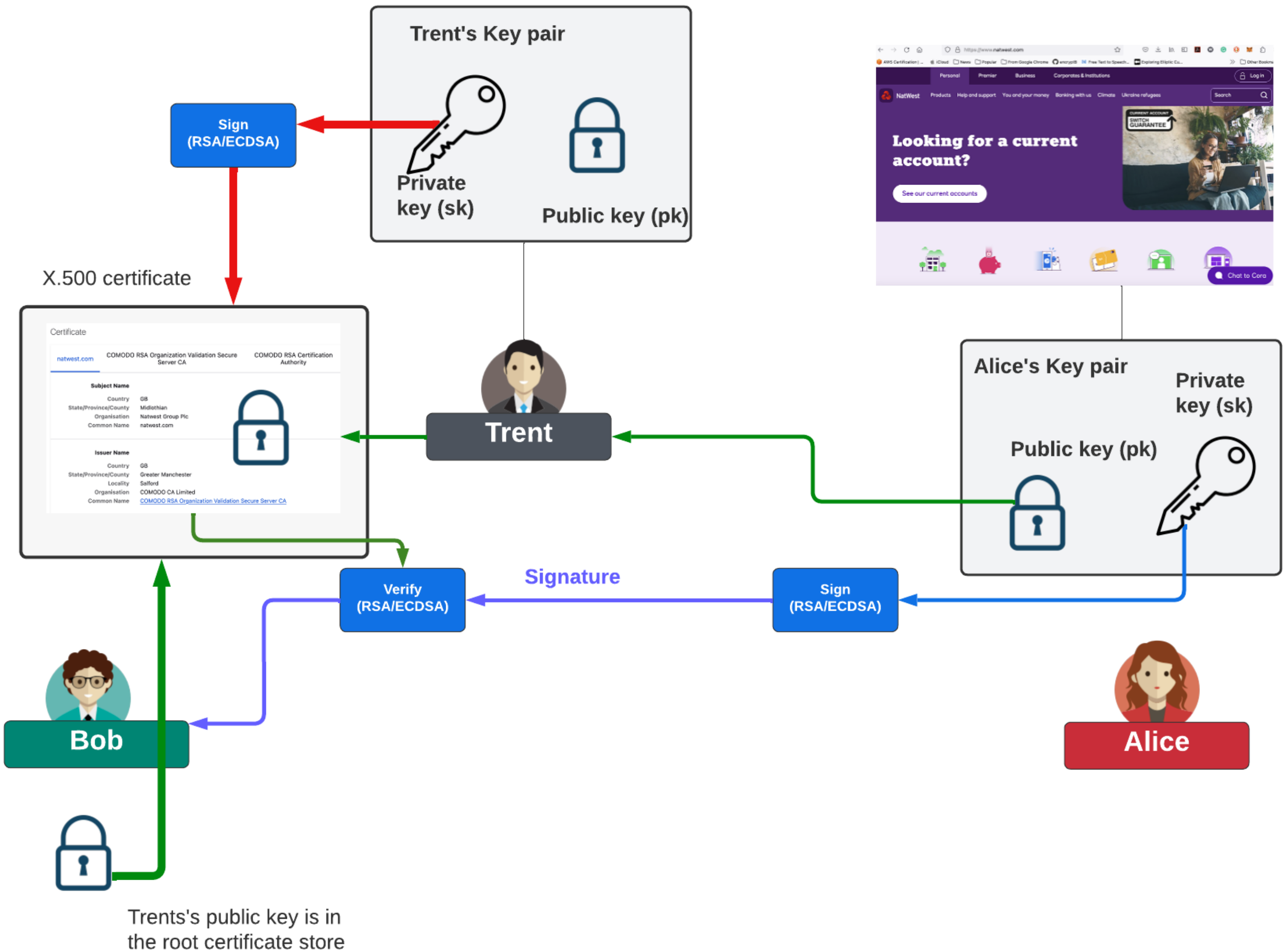
Public Key Encryption



Public Key Digital Signing



Public Key Digital Signing



Public Key

Basics
RSA
Applications (Encryption and Signing)

Prof Bill Buchanan OBE

https://asecuritysite.com/rsa
https://asecuritysite.com/ecc
https://asecuritysite.com/elgamal