Identifier-Based Encryption (IBE)

Trusted Systems Laboratory
February 2003

Identifier-Based Encryption

also known as

Identifier-based Public Key Cryptography
(ID-PKC)

Identifier-Based Encryption

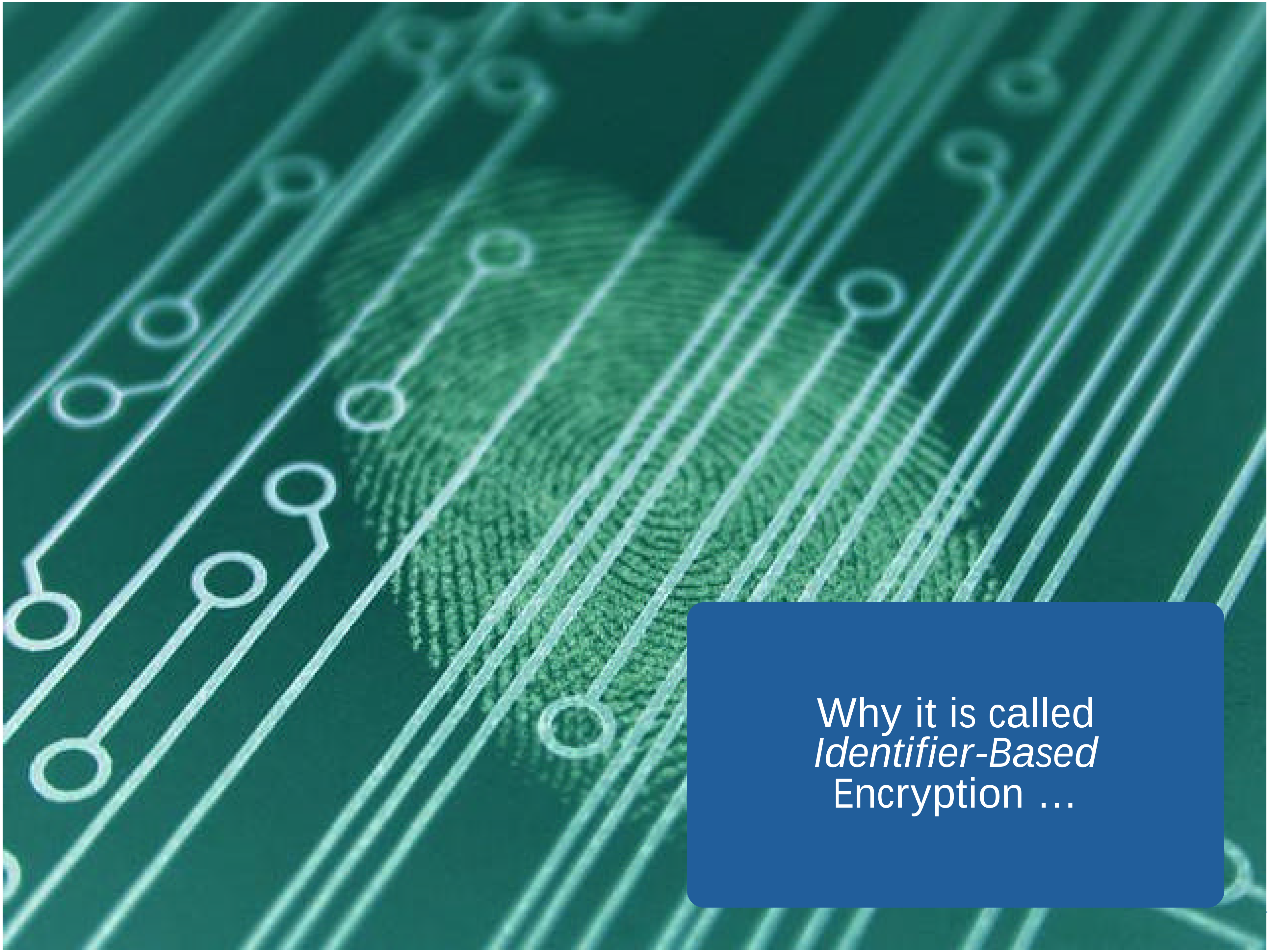
We’ll cover :

- What it is
- Why it is called Identifier-Based Encryption
- Why it is useful
- How it works
- Who is involved with this technology
- Where to find more information

What it is …

Identifier-Based Encryption

- An encryption and decryption technology, based on the use of a public key and a private key
- The public key can be any bit string, which may be chosen by the encrypting party
- The corresponding private key can only be generated by a trusted third party – this need not be done at the same time as the public key is chosen
- The trusted third party controls the release of the private key, so can limit its distribution to those parties who provide evidence of their right to have it
- Parties who are issued with the private key can use it to decrypt content that was encrypted with the public key



Why it is called
Identifier-Based
Encryption …

Identifier-Based Encryption

The encryption key is typically a bit string that identifies the intended decrypting party by a means, and to levels of detail and uniqueness, that are chosen by the encrypting party, and so meet *that* party's requirements for identification of the intended decrypting party.

Why it is useful …

Identifier-Based Encryption

- Encryption schemes solve the problem of keeping data secret from undesired viewers, but create a problem of managing the keys

- With traditional public key cryptography, the generation of the keys, the publication of the associations between parties and their public keys and the management of all this require a dedicated secure infrastructure

- Such an infrastructure is expensive, complex, does not scale well to large sizes, and does not easily extend to manage parties’ attributes, e.g., their roles and rights

- Such issues have not been solved, despite years of attempts – this has limited the take-up of public key cryptography

- IBE offers a much simpler solution for many applications

Identifier-Based Encryption

- Because the encryption key can be any bit string, it can be chosen by the encrypting party. The encrypting party can base its choice of bit string on the needs of the application, such as:
  - something simple, for example
    - the email address of the receiving party
    - a digital photograph of the receiving party
    - a URL
  - something more complex, for example
    - the role of the receiving party, expressed by a list of attributes he must have
    - a set of conditions the receiving party must meet
    - a policy that the receiving party must conform to
    - executable program code
- Expressing these requirements in the encryption key relieves the supporting infrastructure from managing them, thus enabling the infrastructure to be simpler

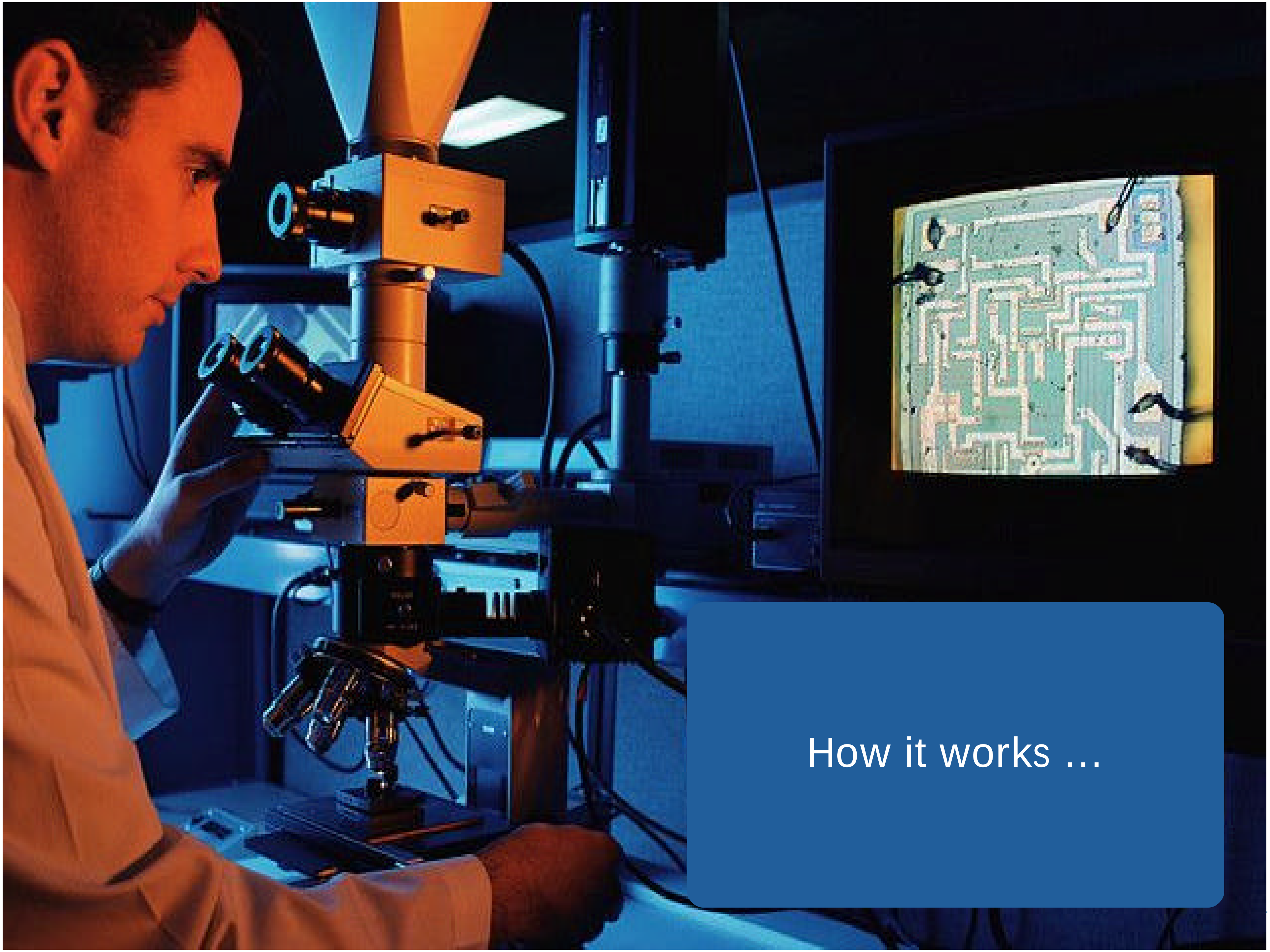
Identifier-Based Encryption

- Because the decryption key does not need to be generated at the same time, or by the same entity, as the encryption key, the trusted third party can delay generating it until the receiving party has demonstrated its right to have it
- So, there is no need for any party to store keys, thus
  - easing the management problem considerably
  - reducing the risk of inadvertently exposed keys compromising the secrecy of the protected content
- A different key can be used, if desired, for every interaction between the sending and receiving parties

Identifier-Based Encryption

It can be used for :

- obscuring content by reversibly scrambling it, thus securing it against viewing by unauthorised parties
- signing content, thus making unrepudiable the link between the content and the signing party
- requiring the consent of multiple third parties (this consent being given, for example, only if the receiving party meets multiple sets of conditions, at least one per third party) before decryption is possible



How it works …

Identifier-Based Encryption

Let’s look at this at two levels :

- operation
- the underlying mathematics

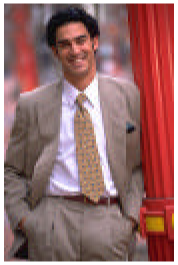
Identifier-Based Encryption - operation

Top-level description – using a simple example

- Three parties:

- Alice, sending a confidential email message to Bob

- Bob, who should be able to read it only if he meets Alice's conditions, in this case that he is a partner in a particular law firm



- Trent, the trusted third party, operated by The Law Society

Identifier-Based Encryption - operation

Some time in the past …

- The Law Society decided to set up a Trusted Third Party (TTP) service to provide obfuscation and signing facilities to its members (the lawyers) and their clients.

- It commissioned the system (Trent) and published:

  - the encryption and decryption algorithms
  - unique system values, P and P_{pub}
  - details of how to access the service (URLs, the syntax and semantics of how to make service requests and express conditions, etc.)

Identifier-Based Encryption - operation

When Alice is ready to send her message to Bob …

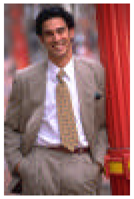
- She obtains from The Law Society website :
  - the encryption algorithm,
  - the system values P and P_{pub},
  - details of how to express the conditions that must be met by Bob before Trent will release the decryption key to him,
  - an applet that manages the process of encrypting and sending the message

- Then she runs the applet

Identifier-Based Encryption - operation

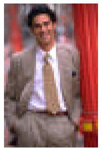
The applet runs in a JVM in Alice's browser. It …

- accepts, via its GUI, her input of the conditions; in this example, she requires that Bob be a partner in the law firm Fox, Weasel & Co
- generates a text string which expresses them
- uses that string as the key to encrypt the message
- packages the encrypted message, the key text string and Trent's access details together
- mails that package to Bob



The applet could also be implemented, for example, as an add-in to her email client, e.g., Outlook, which would remove the need to download it prior to each use.

Identifier-Based Encryption – operation

Bob's  email client receives the package from Alice.

An add-in to it …

- detects that the package contains an encrypted message (together with the encryption key string and TTP details)
- identifies that Trent is the trusted third party
- requests Bob, via a pop-up box, to input some information that will uniquely identify him to Trent

The nature of this identifying information depends on the strength of authentication required by The Law Society. This could range from a simple PIN to a combination of a PIN, a smartcard and some biometrics.

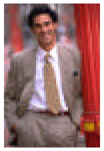
Identifier-Based Encryption - operation

The add in …

- puts Bob’s identifying input and the encryption key string that came in the package from Alice into another package
- and then sends that package to Trent

Identifier-Based Encryption - operation

Trent …

- receives the package sent by Bob’s email client
- decodes the encryption string to obtain the conditions that were specified by Alice
- checks that this package really did come from Bob, by checking the identifying information Bob put in there against its authentication database
- compares Bob’s attributes, stored in its database, against the conditions that were specified by Alice, i.e., that he really is a partner in the law firm Fox, Weasel & Co
- if there is a match, computes the decryption key that corresponds to Alice’s encryption key
- sends that to Bob 

Identifier-Based Encryption - operation

Bob’s email client’s add-in …

- receives the decryption key from Trent
- uses it to decrypt the message from Alice

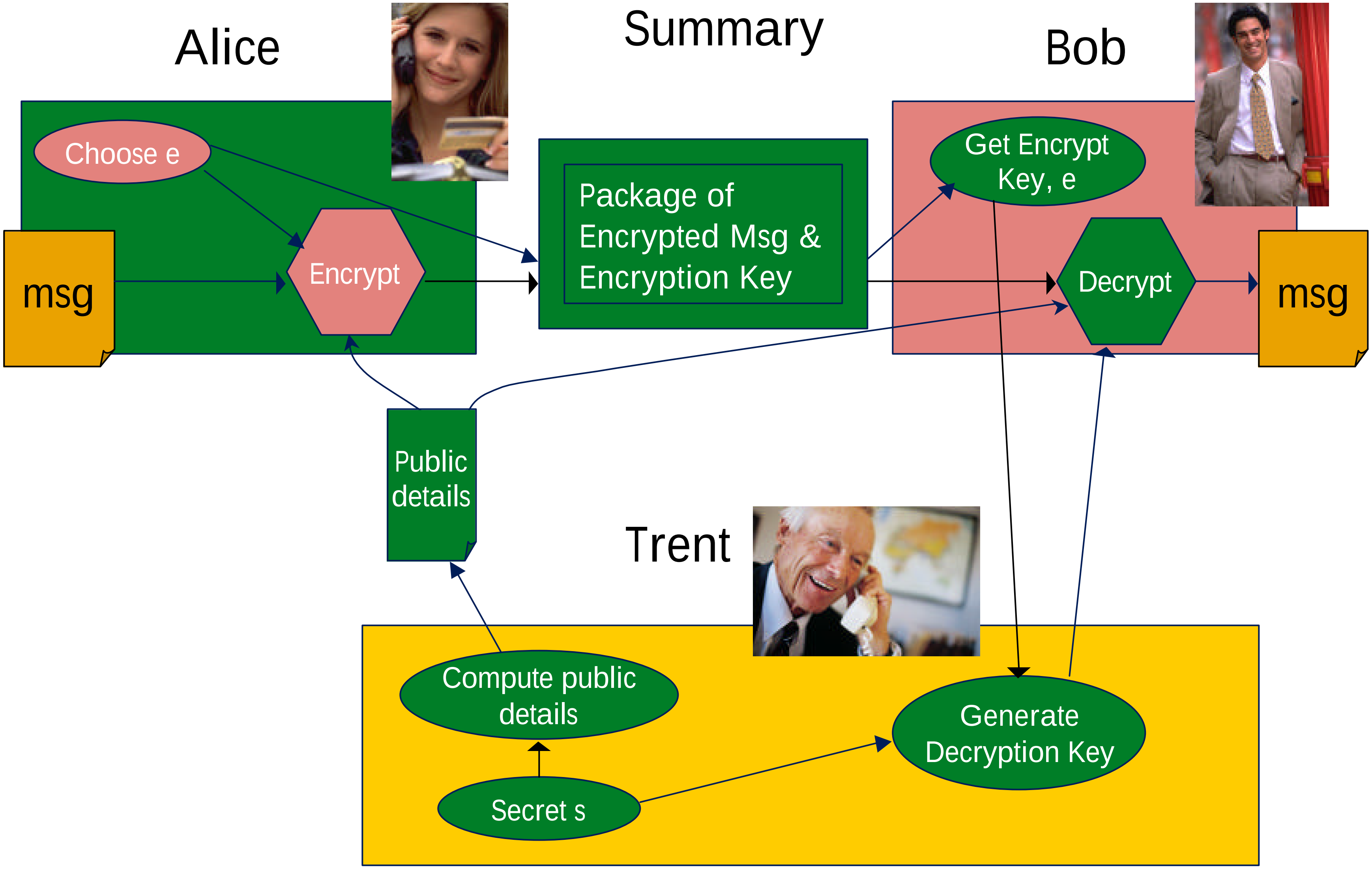
The communication channel between Bob’s email client and Trent must be secure, otherwise eavesdroppers could obtain the decryption key to Alice’s message. There are a number of ways this secure channel can be provided, e.g., by using SSL or https.

Identifier-Based Encryption - operation

Points to note :

- Alice can choose different conditions for each message. For example, she could, if Trent supports an embargo service, also specify that Trent should not release the decryption key until a given time and date
- No keys are stored anywhere, anytime
- There is no need for Alice to know Bob’s name, just his email address – she could choose to send her message to helpdesk@foxweasel.co.uk and just rely on the “partner only” condition to ensure it is read only by a suitable person
- Trent can choose the nature and amount of identifying information Bob must provide, depending on the service level being offered

Identifier-Based Encryption - operation



Alice

Summary

Bob

Choose e

msg

Encrypt

Package of Encrypted Msg & Encryption Key

Get Encrypt Key, e

Decrypt

msg

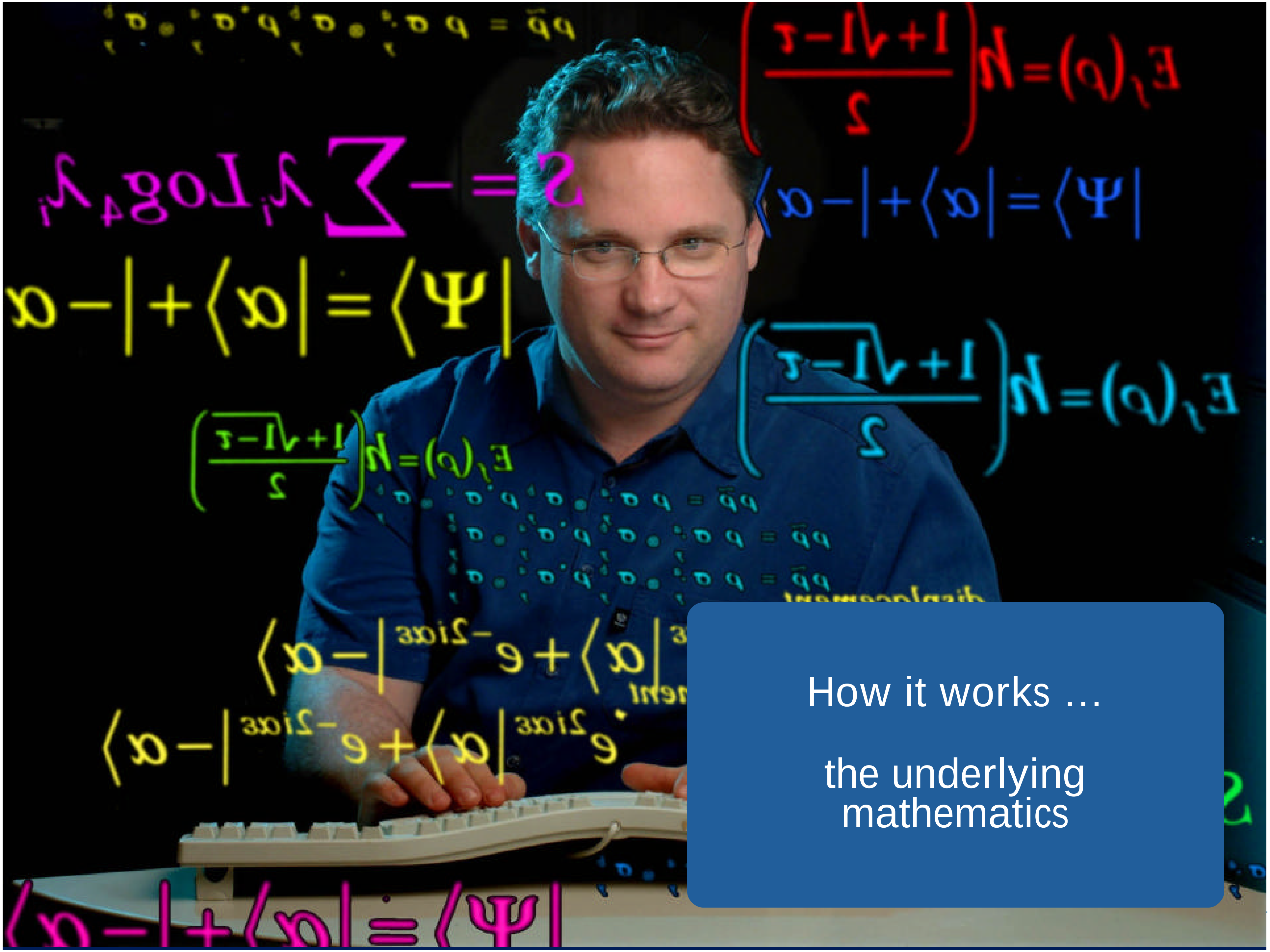
Public details

Trent

Compute public details

Generate Decryption Key

Secret s



How it works …

the underlying mathematics

Identifier-Based Encryption - mathematics

There are two completely different ways of implementing IBE :

- using Quadratic Residuosity; this was invented by Clifford Cocks of CESG (www.cesg.gov.uk)
- using Elliptic curves and Pairing functions; this was invented by Dan Boneh and Matt Franklin at Stanford University

The Pairing-based solution has proven to be the more versatile of the two; this is the approach we have taken.

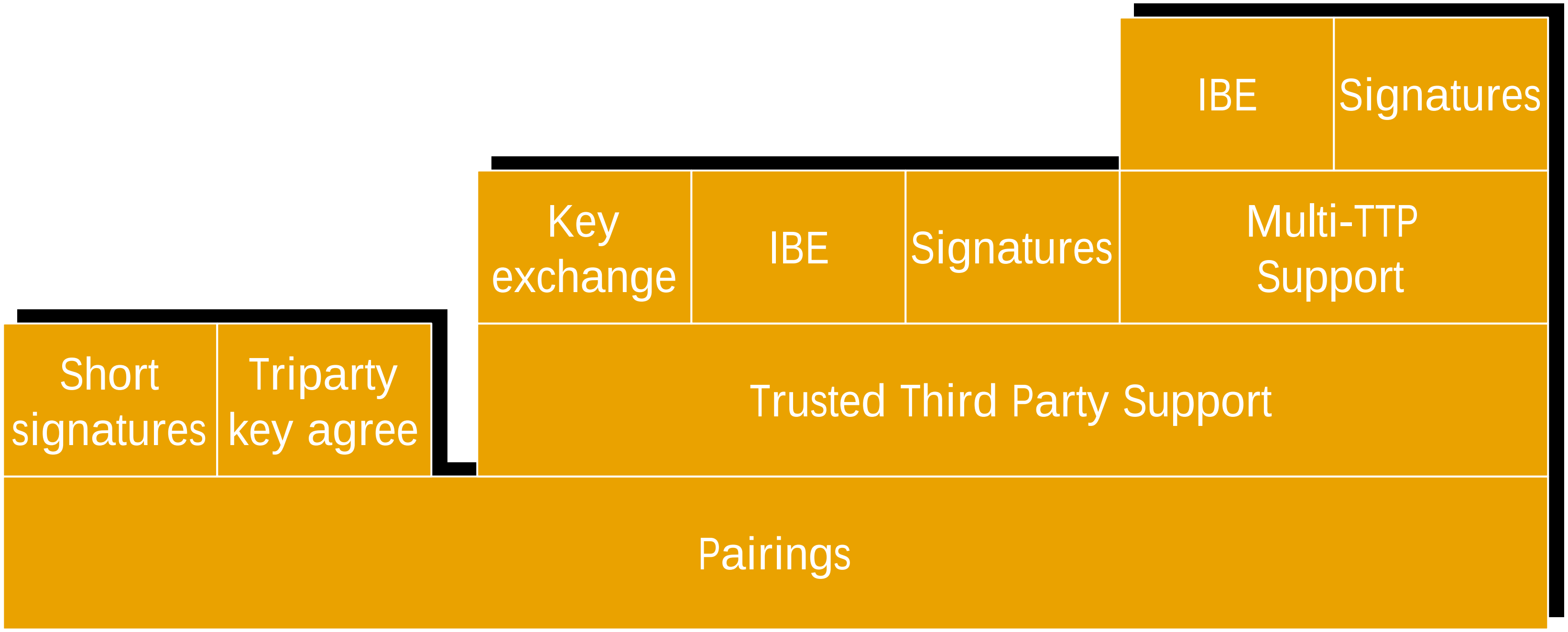
Identifier-Based Encryption - mathematics

- A Pairing is a bilinear mapping
  - $\hat{e}(P1, P2) \Rightarrow R$ where P1 and P2 are points on an elliptic curve and R is an integer mod p^e
- It has the following properties
  - $\hat{e}([k]P, Q) = \hat{e}(P, Q)^k$
  - $\hat{e}(P, [m]Q) = \hat{e}(P, Q)^m$
  - $\hat{e}([k]P, [m]Q) = \hat{e}(P, Q)^{km}$
  - $\hat{e}(P, P) \neq 1$ for most points P
- Only a few bilinear mappings are known. These include the Weil and Tate Pairings.

Note: $[k]P = P + P + \ldots + P$, where there are k summands.

Identifier-Based Encryption - mathematics

Pairings are incredibly useful



Identifier-Based Encryption - mathematics

Pairing algorithms for IBE (1)

To set up Trent:

1. Choose point P (probably an industry-wide standard value)
2. Choose random secret integer s
3. $P_{pub} = [s]P$
4. Publish P and P_{pub}

To encrypt and send a message:

1. Choose encryption key, str
2. Q_{id} = convertStringToPoint(Hash1(str))
3. Choose random r
4. Compute $U = [r]P$
5. Compute $g_{ID} = \hat{e}(Q_{id}, P_{pub})$
6. Compute otp = Hash2($g_{ID}{}^{r}$)
7. Compute V = m XOR otp
8. Package & ship (U, V)

Identifier-Based Encryption - mathematics

Pairing algorithms for IBE (2)

Trent generates the decryption key:

1. Q_{id} = convertStringToPoint(Hash1(str))
2. $d_{ID} = [s]Q_{id}$
3. Return d_{ID}

To decrypt (U,V):

1. Compute $x = \hat{e}(d_{ID}, U)$
2. Compute otp = Hash2(x)
3. Compute m = V xor otp

Identifier-Based Encryption - mathematics

A few principles that underpin the algorithms :

- It is assumed that s is secret and known only to Trent. It is assumed to be not possible to compute s given P and P_{pub}. Similarly, there is no way of computing s using a matching encryption/decryption key pair.

- Using the secret information s, Trent is easily able to compute the decryption key corresponding to a supplied encryption key string.

- There is no known feasible way of deriving the decryption key without knowing s.

Who is involved?

Identifier-Based Encryption

The Trusted Systems Laboratory within Hewlett-Packard Laboratories in Bristol, England, started work on IBE in 2001, and has developed a library of cryptographic software and various demonstrators.

HP is also collaborating with researchers at the University of Bristol and Royal Holloway University of London.

Identifier-Based Encryption

HP and CESG jointly ran a demonstration of IBE at the Royal Society Summer Exhibition in London in June 2002.

HP has implemented a prototype secure role-based email system, based on IBE, for a live trial in a healthcare application.

Where to find more information …

Identifier-Based Encryption

Chen, L., K. Harrison, A. Moss, D. Soldera, N.P. Smart, "Certification of Public Keys within an Identity Based System", Proc. 5th International Information Security Conference (ISC), 2002. LNCS 2433, Springer-Verlag, 2002.

Keith.Harrison@hp.com

hp®

invent