Email Security

Email Security

- E-mail is one of the most widely used network services
  - One of the old killer applications of the Internet
- Normally message contents not secured
  - Can be read/modified either in transit or at destination by the attacker
- E-mail service is like postcard service
  - just pick it and read it

Email Security Enhancements

- confidentiality
  - protection from disclosure
- authentication
  - of sender of message
- message integrity
  - protection from modification
- non-repudiation of origin
  - protection from denial by sender

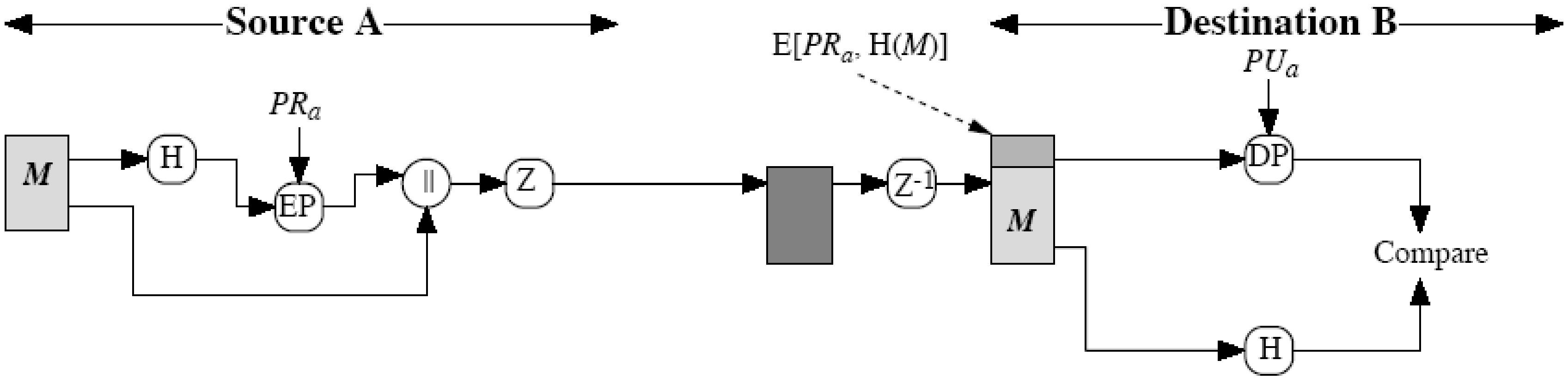
Pretty Good Privacy (PGP)

- Widely used secure e-mail software
  - originally a file encryption/decryption facility
- Developed by Phil Zimmermann
- Best available crypto algorithms are employed
- Available on several platforms with source code
- Originally free, now commercial versions exist
- Not controlled by a standardization body
  - although there are RFCs

PGP Mechanisms

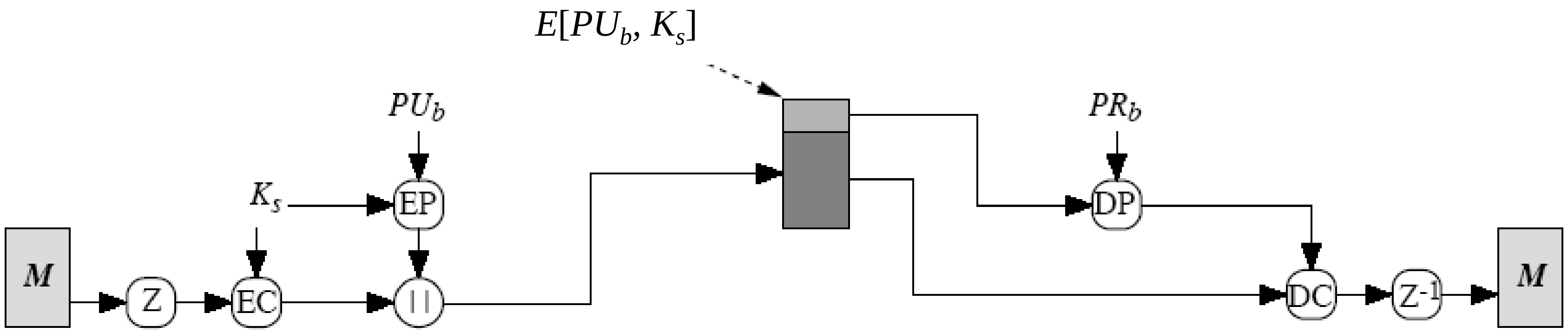
- Digital Signatures (and consequently message authentication and integrity)
  - RSA, DSS, and others
- Message Encryption
  - CAST, IDEA, 3DES, AES, etc. (all at least 128 bits)
  - symmetric keys are used once and encrypted using RSA or ElGamal
- Compression using ZIP
- Radix-64 conversion (to ASCII)
  - for e-mail compatibility

PGP Operation – Digital Signatures



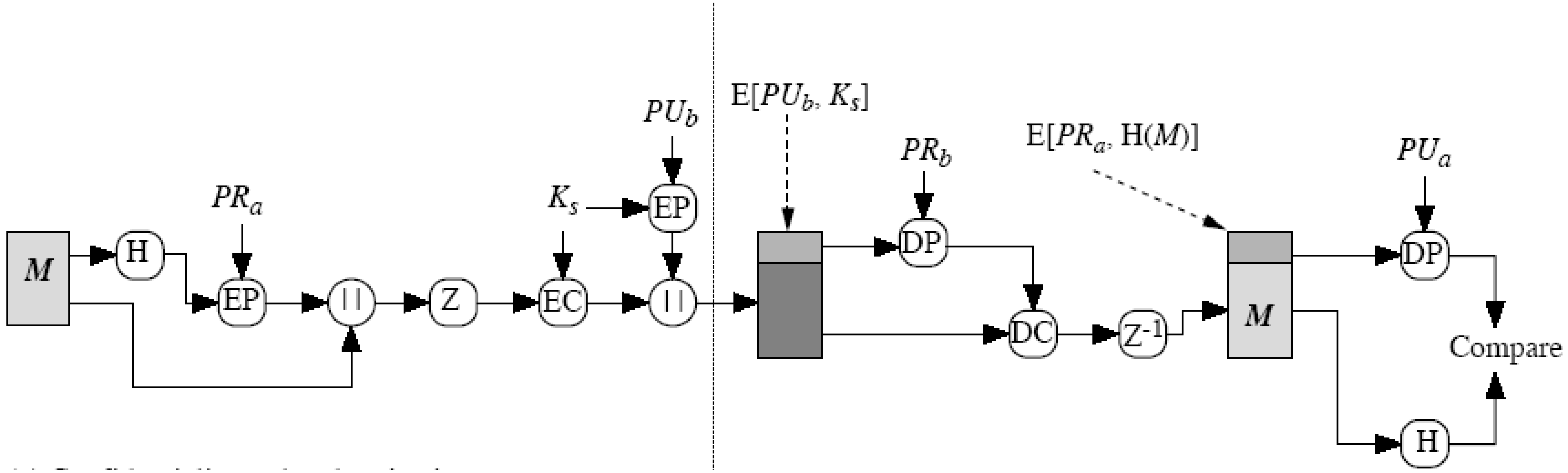
- Classical application of public key crypto
  - This figure is actually for RSA
  - for DSA refer to previous lectures
- Z is zip function
- radix-64 conversion is done after zip at sender, before Z^{-1} at receiver
  - may be done only for signature or for the whole message

PGP Operation – Confidentiality



- One-time session key, K_s
  - generated at random
  - encrypted using a public key cryptosystem, EP
    - RSA or ElGamal
- Message is compressed before encryption
  - This is the default case

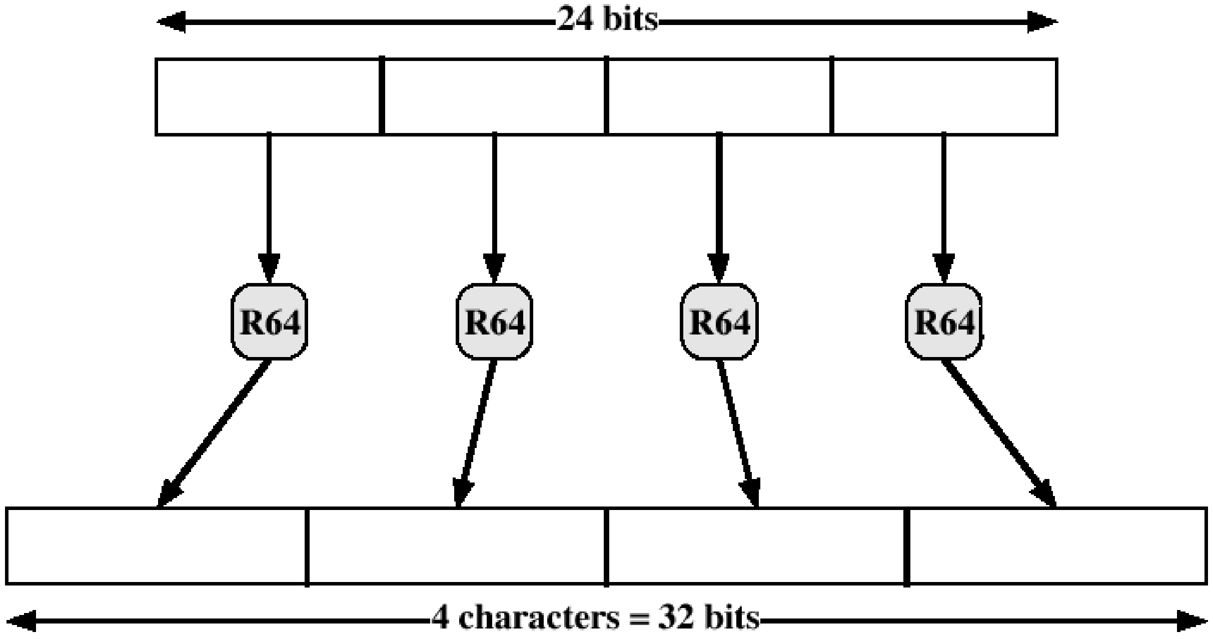
PGP Operation – Confidentiality and Authentication



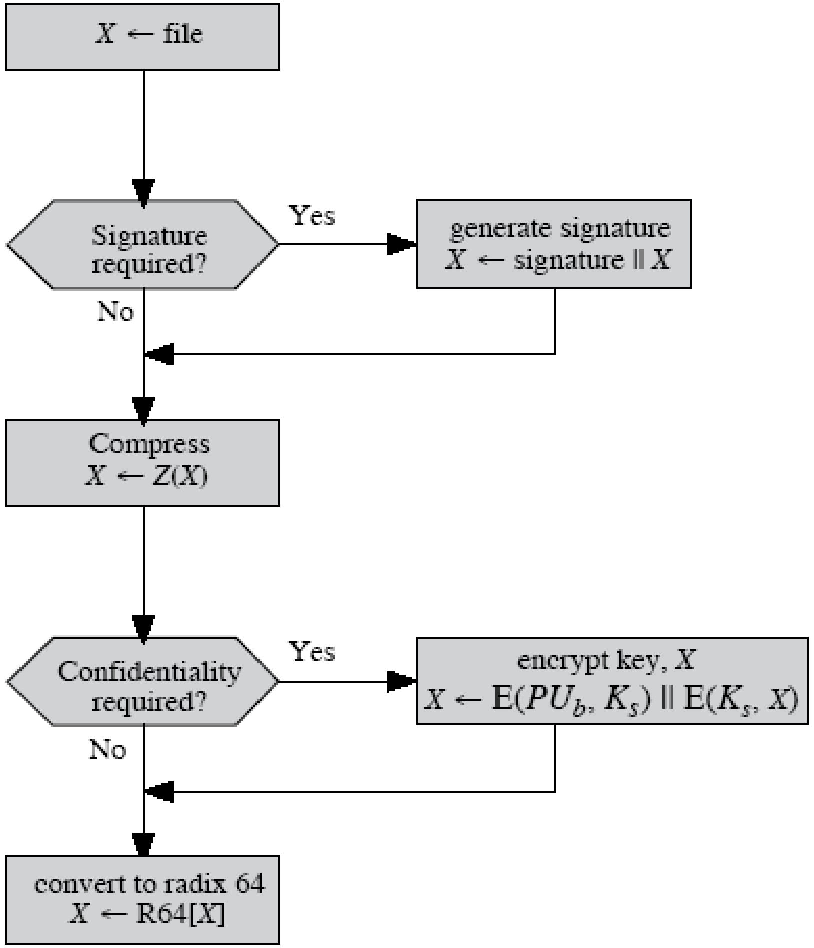
- uses both services on same message
  - create signature and attach to message
  - compress and encrypt both message & signature
  - attach encrypted session key
  - radix-64 conversion is for everything at the end

PGP Operation – radix-64 conversion

- Encrypted text and signatures create binary output
- however email was designed only for text
  - hence PGP must encode raw binary data into printable ASCII characters
- uses radix-64 algorithm (See CS408 notes)
  - maps 3 bytes to 4 printable chars

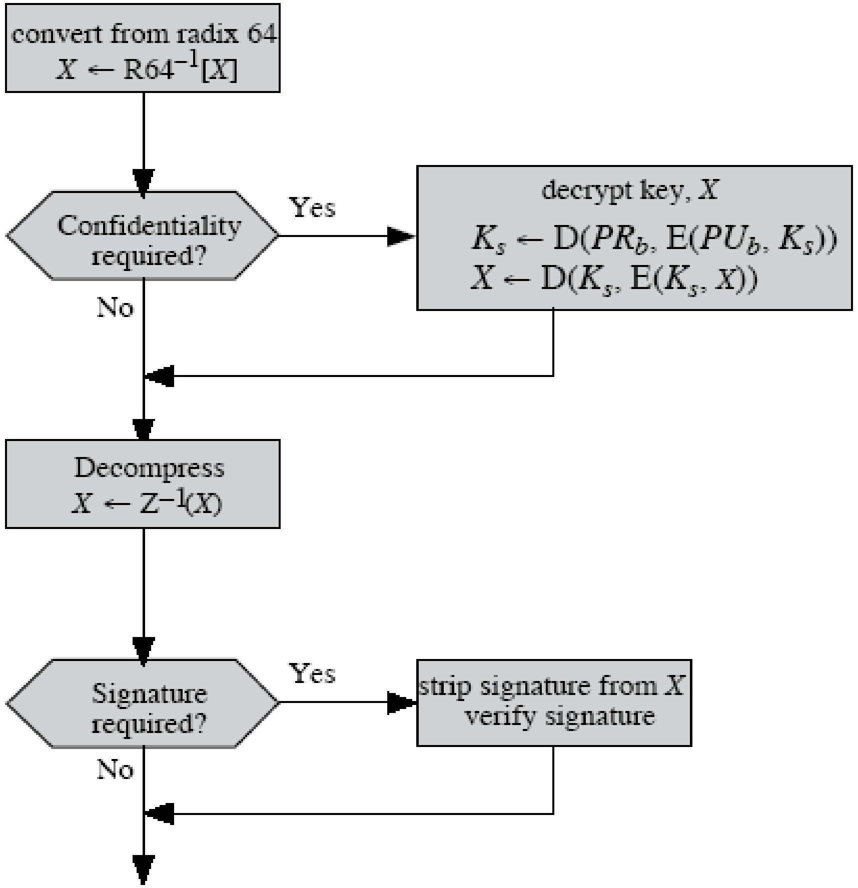


PGP Operation – Summary



$X \leftarrow$ file
Signature required?
Yes
generate signature
$X \leftarrow$ signature $\|$ X
No
Compress
$X \leftarrow Z(X)$
Confidentiality required?
Yes
encrypt key, X
$X \leftarrow \mathrm{E}(PU_b, K_s) \| \mathrm{E}(K_s, X)$
No
convert to radix 64
$X \leftarrow \mathrm{R64}[X]$

(a) Generic Transmission Diagram (from A)



convert from radix 64
$X \leftarrow \mathrm{R64}^{-1}[X]$
Confidentiality required?
Yes
decrypt key, X
$K_s \leftarrow \mathrm{D}(PR_b, \mathrm{E}(PU_b, K_s))$
$X \leftarrow \mathrm{D}(K_s, \mathrm{E}(K_s, X))$
No
Decompress
$X \leftarrow Z^{-1}(X)$
Signature required?
Yes
strip signature from X
verify signature
No

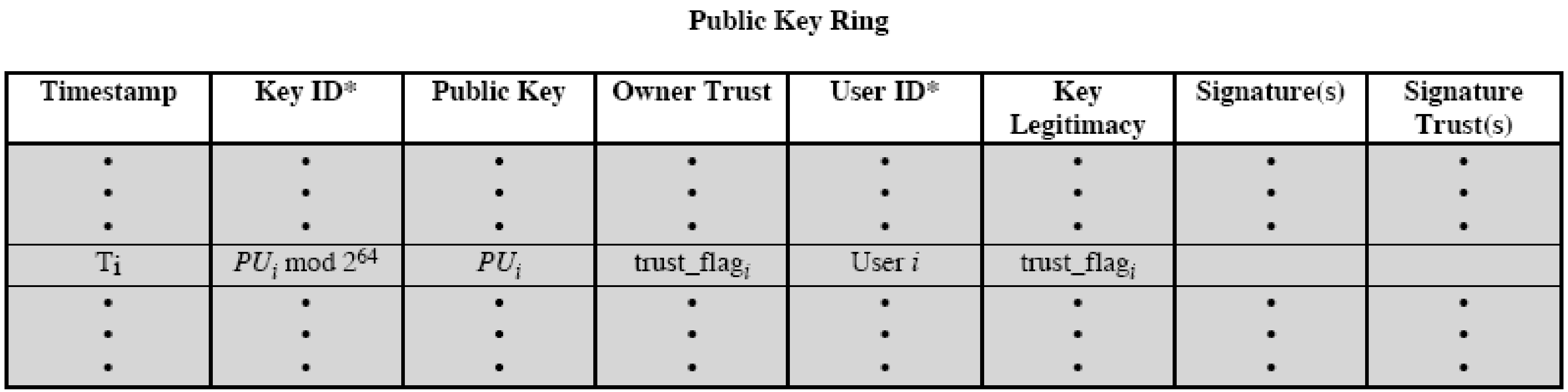
(b) Generic Reception Diagram (to B)

PGP Key ID concept

- since a user may have many public/private keys in use, there is a need to identify which is actually used to encrypt the session key in a message
    - PGP uses a key identifier which is least significant 64-bits of the public key
    - uniqueness?
        - very likely, at least for a particular user ID (e-mail address)
- Key IDs are used in signatures too
    - key ID for the public key corresponding to the private key used for signature
- Key IDs are sent together with messages

PGP Key Rings

- each PGP user has a pair of keyrings to store public and private keys
  - public-key ring contains all the public-keys of other PGP users known to this user



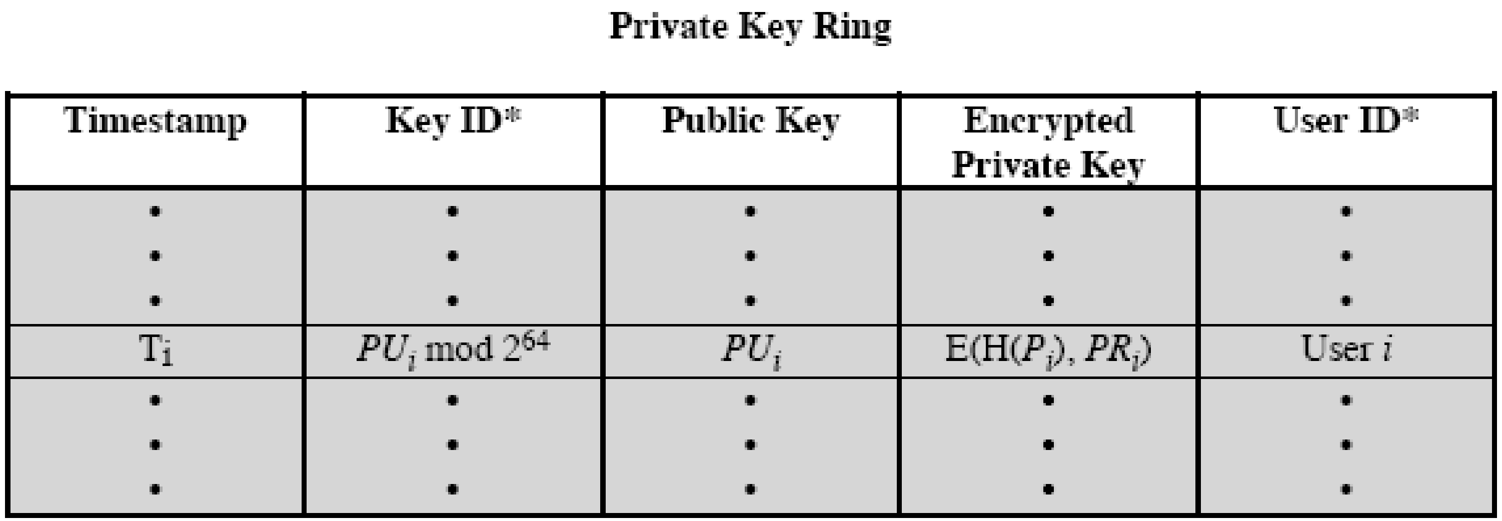
Public Key Ring

Timestamp	Key ID*	Public Key	Owner Trust	User ID*	Key Legitimacy	Signature(s)	Signature Trust(s)
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •
T_i	PU_i mod 2^{64}	PU_i	trust_flag$_i$	User i	trust_flag$_i$		
• • •	• • •	• • •	• • •	• • •	• • •	• • •	• • •

* Key ID, Used ID combination uniquely identifies a key

PGP Key Rings

- private-key ring contains the public/private key pair(s) for this user,
- private keys are encrypted using a key derived from a hashed passphrase

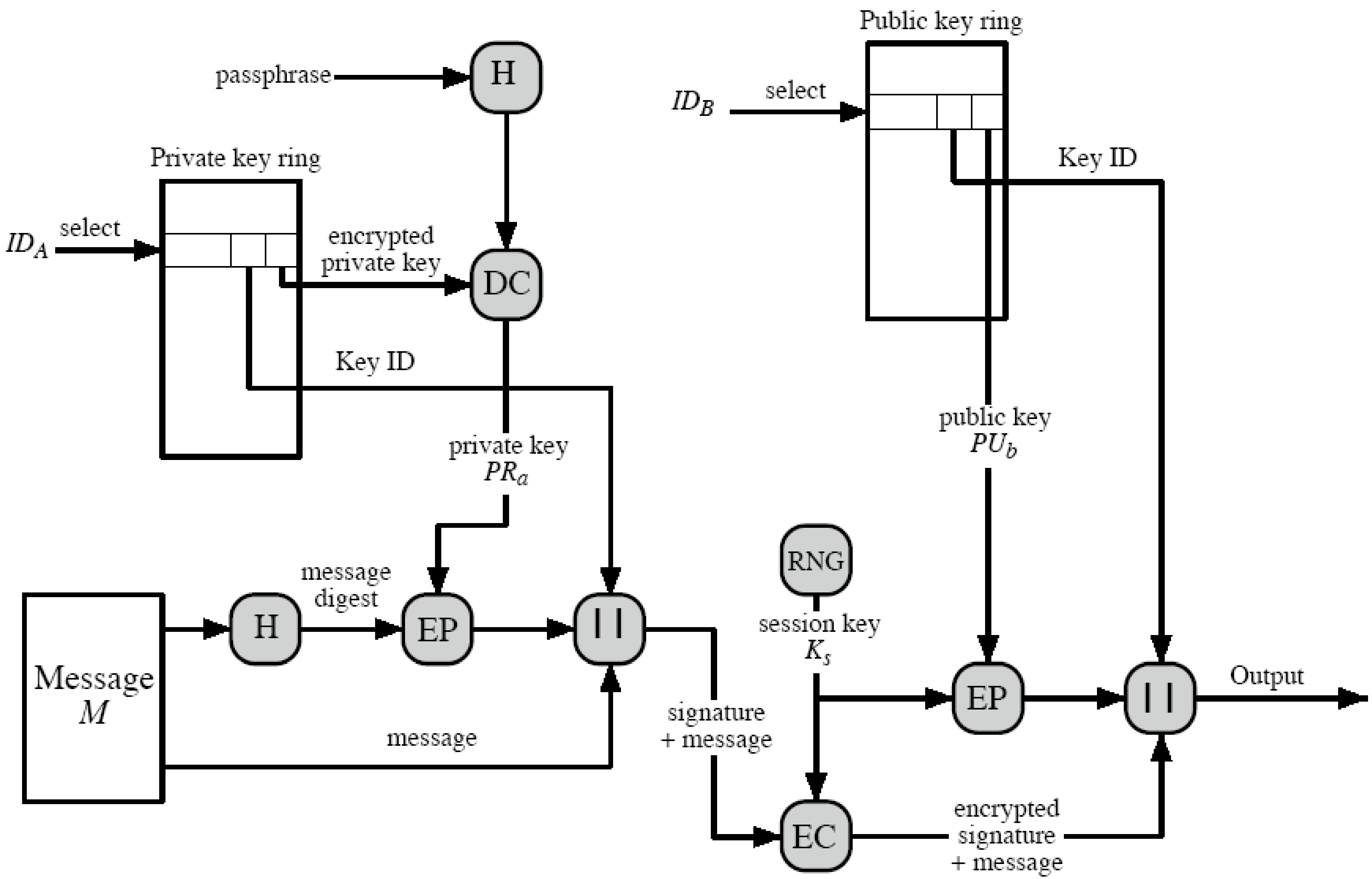


Private Key Ring

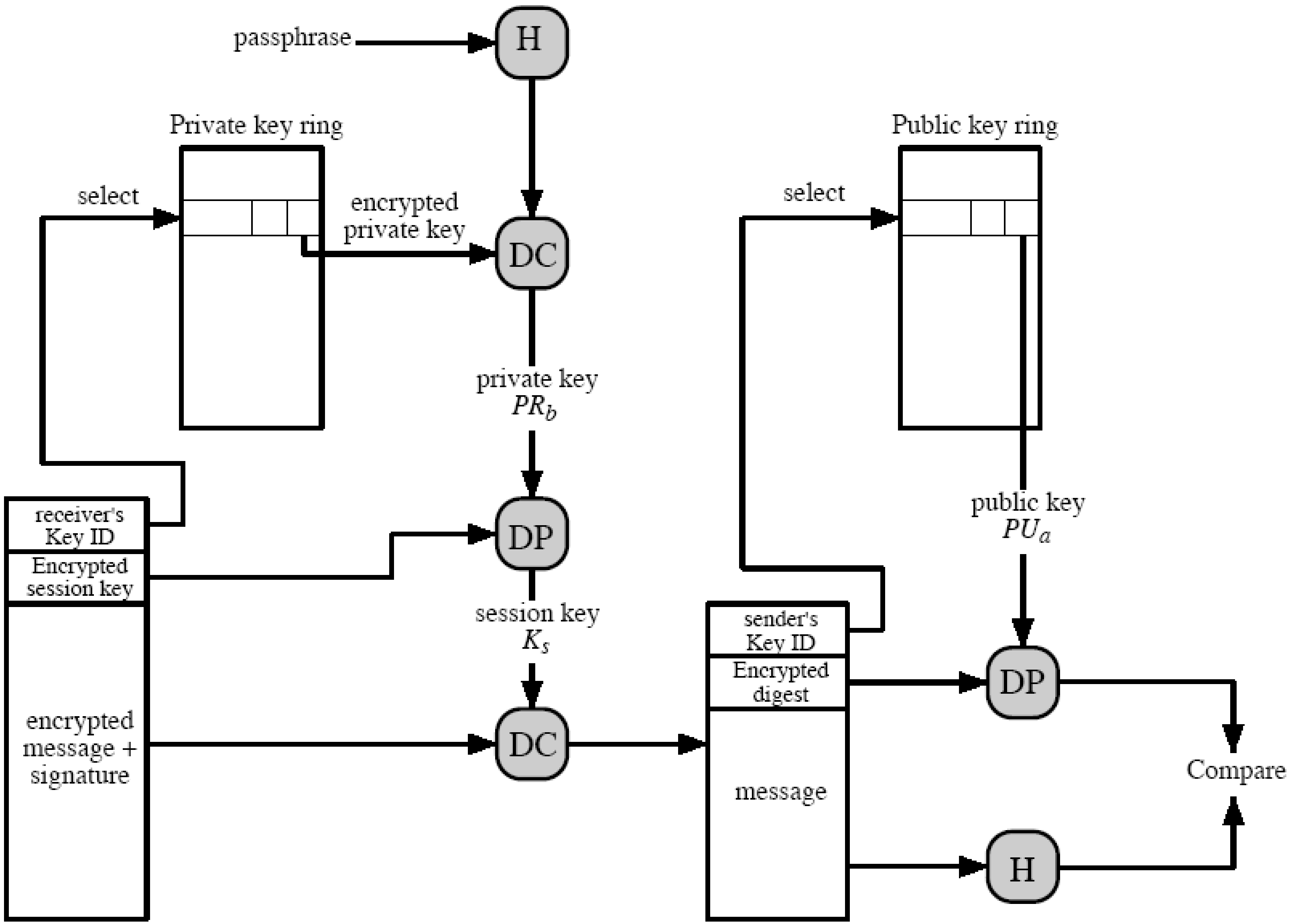
Timestamp	Key ID*	Public Key	Encrypted Private Key	User ID*
• • •	• • •	• • •	• • •	• • •
T_i	$PU_i \bmod 2^{64}$	PU_i	$E(H(P_i), PR_i)$	User i
• • •	• • •	• • •	• • •	• • •

[*] Key ID, Used ID combination uniquely identifies a key

Key rings and message generation



Key rings and message reception



Quick E-mail History

- SMTP and RFC 822 (later RFC 5322)
  - SMTP is the email transfer protocol running over TCP
  - RFC 822/5322 defines the message format and headers
    - only ASCII messages (7-bit)
- MIME (Multipurpose Internet Mail Extensions)
  - content type
    - Almost any type of information can appear in an email message
  - transfer encoding
    - specifies how the message body is encoded into textual form (radix64 is common)
- S/MIME: Secure MIME
  - new content types, like signature, encrypted data

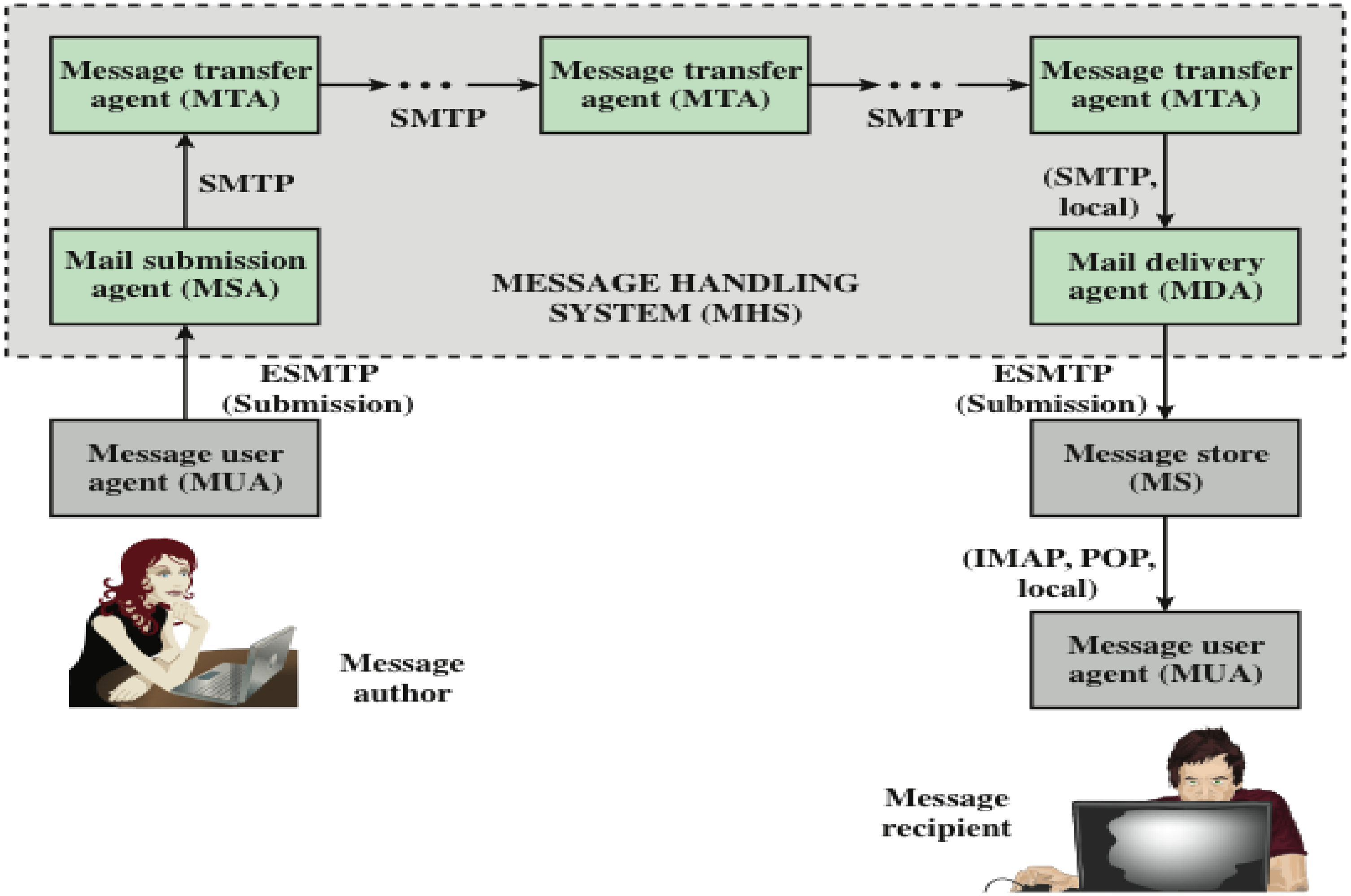
Email Terminologies

- Mail User Agent (MUA): a program which provides a human user interface for reading and sending mail.
  - Examples: elm, pine, mutt, Outlook, Netscape, Thunderbird.
- Mail Transport Agent (MTA): a program which acts as a "mail server". Specifically, it's responsible for managing a queue of outgoing mail, and for accepting (or rejecting) incoming mail.
  - Examples: sendmail, qmail, postfix, exim.
- Mail Submission Agent (MSA): a relatively new term in the e-mail field.
  - This is the component of an MTA which accepts new mail messages from an MUA, using SMTP.

Continued..

- Mail Delivery Agent (MDA): the component of an MTA which is responsible for the final delivery of a message to a local mailbox on disk.
    - Sometimes this is a separate program, and sometimes it's built into the MTA
- Post Office Protocol (POP): a protocol used by some MUAs to retrieve mail from a user's mailbox on a remote server.
    - Often written "POP3".
    - The official TCP port number for POP3 is 110.
- Internet Message Access Protocol (IMAP): a protocol used by some MUAs to retrieve mail from a user's mailbox on a remote server.
    - This is a newer and more complicated protocol than POP, with a lot more functionality.
    - The official TCP port number for IMAP is 143.
    - IMAP is more flexible and complex than POP3.
- The **main difference** is that **IMAP**(Internet Messaged Access Protocol) always syncs with mail server so that any changes you make in your mail client (Microsoft Outlook, Thunderbird) will instantly appear on your webmail inbox.

More on Internet Email Architecture

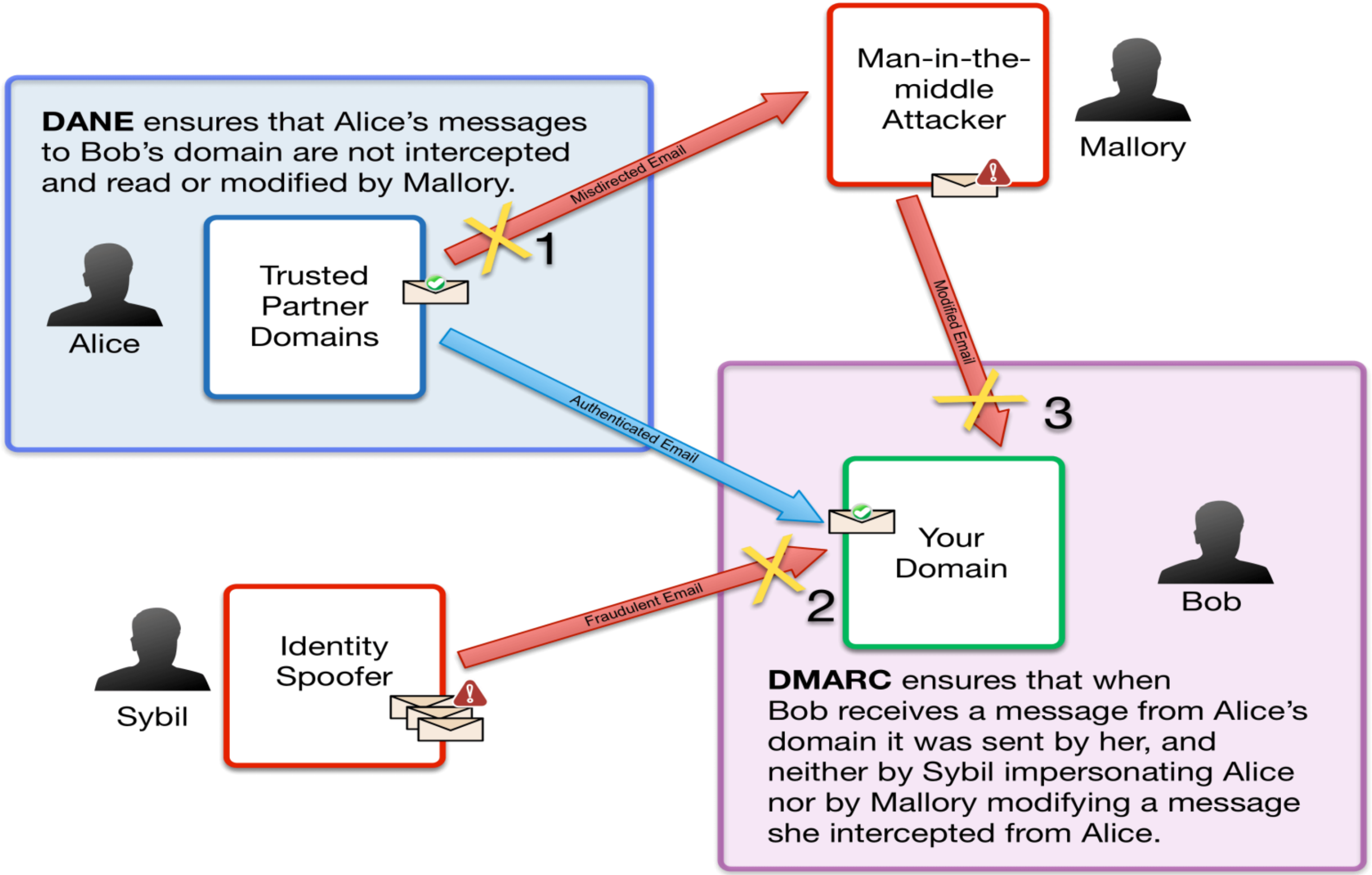


Post Office Protocol (POP3)	Internet Message Access Protocol (IMAP)
POP is a simple protocol that only allows downloading messages from your Inbox to your local computer.	IMAP is much more advanced and allows you the user to see all the folders on the mail server.
The POP server listens on port 110, and the POP with SSL secure(POP3DS) server listens on port 995	he IMAP server listens on port 143, and the IMAP with SSL secure(IMAPDS) server listens on port 993.
In POP3 the mail can only be accessed from a single device at a time.	Messages can be accessed across multiple devices
To read the mail it has to be downloaded on the local system.	The mail content can be read partially before downloading.
The user can not organize mails in the mailbox of the mail server.	The user can organize the emails directly on the mail server.
The user can not create, delete or rename email on the mail server.	The user can create, delete or rename email on the mail server.
A user can not search the content of mail before downloading to the local system.	A user can search the content of mail for specific string before downloading.
After download, the message exists in the local system if the local system crashes message is lost.	Multiple redundant copies of the message are kept at the mail server, in case of loss of message of a local server, the mail can still be retrieved
Changes in the mail can be done using local email software.	Changes made web interface or email software stay in sync with the server.
All the message are downloaded at once.	Message header can be viewed prior to downloading.

DANE and DMARC

- DNS-based Authentication of Named Entities (DANE)
  - DANE, or RFC6698, is intended to mitigate the threat of a man-in-the-middle intercepting encrypted communications by posing as one of the end points.
- DMARC, or RFC7489, is intended to mitigate the threat of an arbitrary sender
- DMARC, which stands for “Domain-based Message Authentication, Reporting & Conformance”
  - an email authentication, policy, and reporting protocol.
  - It builds on the widely deployed SPF and DKIM protocols, adding linkage to the author (“From:”) domain name, published policies for recipient handling of authentication failures, and reporting from receivers to senders, to improve and monitor protection of the domain from fraudulent email.

DANE and DMARC



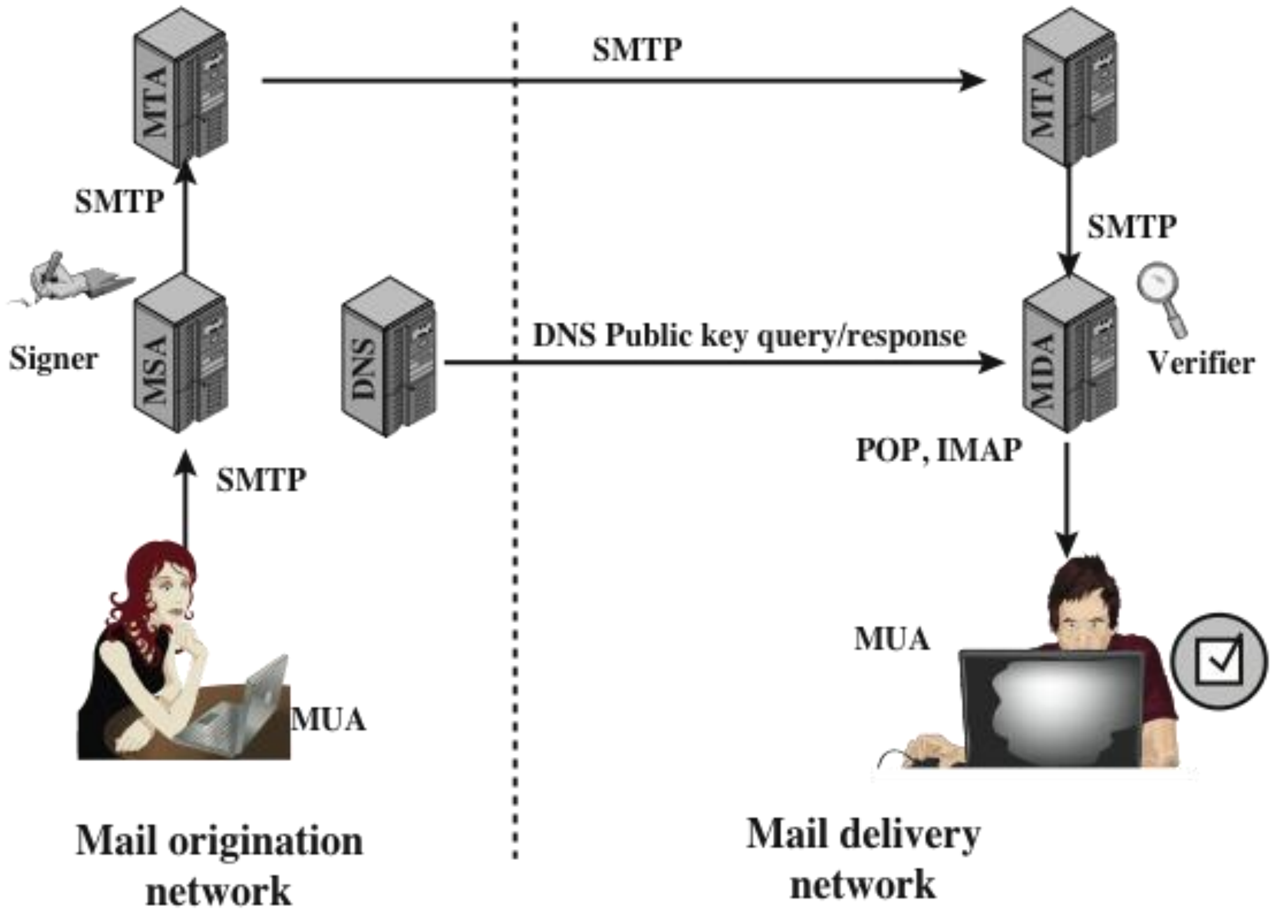
DANE ensures that Alice's messages to Bob's domain are not intercepted and read or modified by Mallory.

DMARC ensures that when Bob receives a message from Alice's domain it was sent by her, and neither by Sybil impersonating Alice nor by Mallory modifying a message she intercepted from Alice.

Courtesy: https://www.agari.com/email-security-blog/dane-vs-dmarc-the-email-authentication-landscape/

DomainKeys Identified Mail (DKIM)

- Started as an industrial effort but later defined in RFC 6376
  - Adopted widely by a range of e-mail providers and Internet Service Providers (ISPs)
- Basically, signing the emails.
  - But not by the sender; but by the sending mail server
  - Similarly, the verifier is not the recipient user, but the receiving mail server.
  - So not end-to-end, but between sending MTA and receiving MTA (or agents on behalf of the MTAs)
- By signing an email,
  - The sending domain (via its MTA or its agent) claim responsibility for the email – it says “my server is sending it”
  - Thwarts server-spoofing attacks cryptographically
  - But it does not provide any proof about the individual who wrote the email
- Public-private key pairs belong to the domains
  - Public keys are stored as DNS records
- Receiving domain (via its MTA or its agent) verifies the signature before passing the email to the ultimate recipient.
  - Public key of the sender is obtained via a DNS query

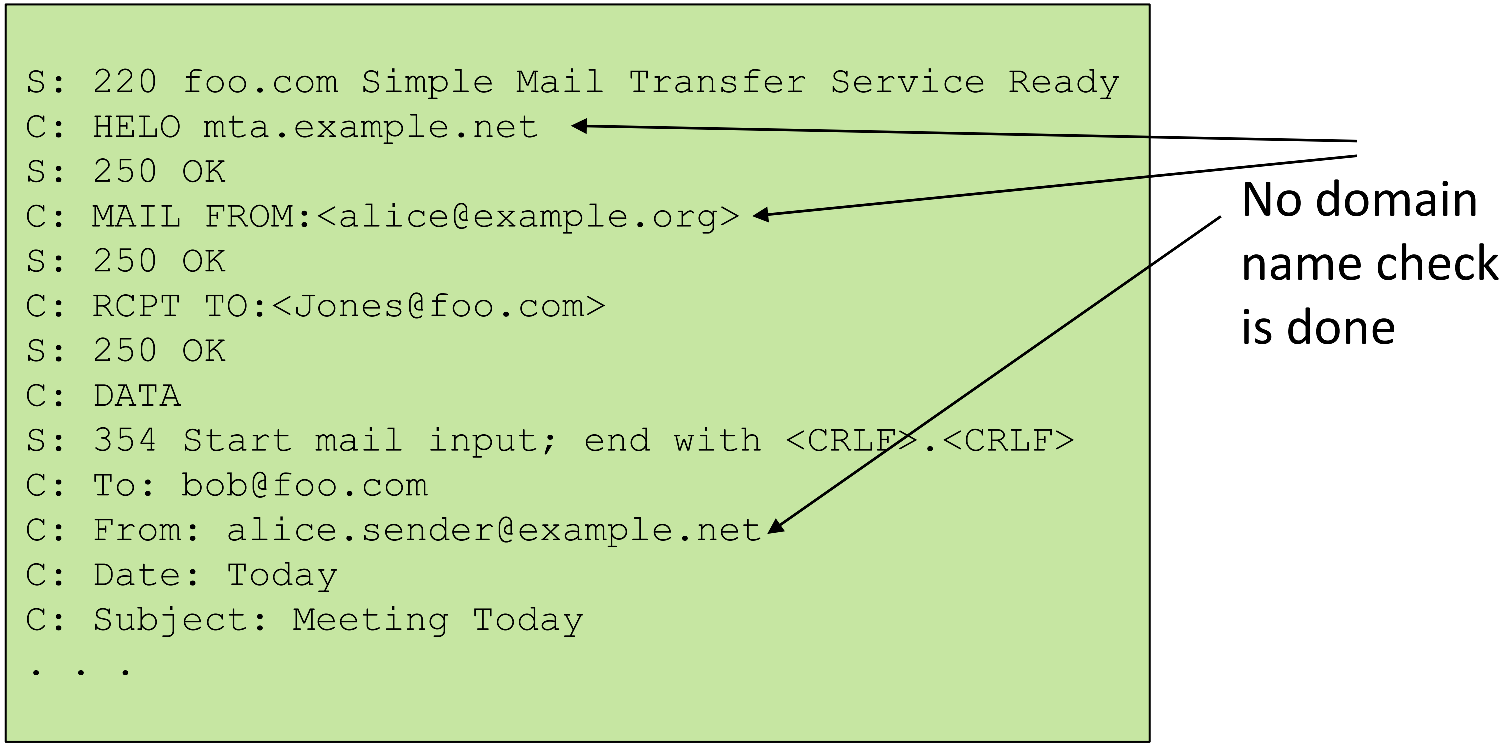


DNS = domain name system
MDA = mail delivery agent
MSA = mail submission agent
MTA = message transfer agent
MUA = message user agent

Figure 19.10 Simple Example of DKIM Deployment

Sender Policy Framework (SPF)

- Current email infrastructure allows to use any domain name while sending during SMTP messaging and in the email header.
  - This is the problem that SPF addresses



```
S: 220 foo.com Simple Mail Transfer Service Ready
C: HELO mta.example.net
S: 250 OK
C: MAIL FROM:<alice@example.org>
S: 250 OK
C: RCPT TO:<Jones@foo.com>
S: 250 OK
C: DATA
S: 354 Start mail input; end with <CRLF>.<CRLF>
C: To: bob@foo.com
C: From: alice.sender@example.net
C: Date: Today
C: Subject: Meeting Today
. . .
```

No domain name check is done

Sender Policy Framework (SPF)

- SPF is the standardized way for a sending domain to identify and assert the mail senders for that domain
- RFC 7208 defines the SPF
  - Domain admins can authorize hosts to use their domain names in “MAIL FROM” or “HELLO” identities during SMTP transactions
- SPF works by checking a sender’s IP address against the policy encoded in any SPF record found at the sending domain DNS records
  - This means that SPF checks can be applied before the message content is received from the sender (from: field in the message header is not checked)
- SPF policies are stored in TXT type DNS resource records (SPF TXT RR)

SPF on the Sender Side

- Basically formation of the policy and adding it to DNS of the sender’s domain
  - Sending domain identifies all the allowed senders
  - Creates a policy using SPF syntax and adds to DNS
- SPF policy is made of “SPF Mechanisms” and “Mechanism Modifiers”
- SPF mechanisms
  - IP addresses (v4 or v6): a list or a range (mostly to allow as sender)
  - MX: to allow the mail servers registered at DNS of that domain as legitimate senders.
  - include: a domain name proceeds “include” and generally this mechanism is used to check other domain’s SPF policies for allowed senders. Generally it is used when the email service is outsourced (e.g. to Microsoft or Google)
  - all: all others
- SPF Mechanism Modifiers
  + mechanism allowed (default one, so if no modifier mentioned, it is +)
  - mechanism not allowed
  ~ soft fail: mechanism allowed but marked as suspicious; triggers extra checks
  ? neutral: SPF does not say anything, system’s default is applied (mostly +)

SPF Policy Examples (not in the book)

- v=spf1 include:spf.protection.outlook.com –all
  - ❑ “v=spf1” means spf version 1; not so important
  - ❑ There is a default + before “include” that means the same policy could have been written as:

    v=spf1 +include:spf.protection.outlook.com –all
  - ❑ include:spf.protection.outlook.com means check Microsoft for the SPF policies; I do not know any ☺
  - ❑ –all means reject the rest
- v=spf1 mx ip4:67.138.237.135 include:spf.protection.outlook.com -all

  mx (could also be written as +mx) means the mail servers in the DNS are allowed

  A particular IP address (67.138.237.135) is allowed

  The rest is the same as previous example

SPF Policy Examples (not in the book)

- Sabancı University’s SPF policy as of April 25, 2019.

```
v=spf1 ip4:193.255.135.0/23 ip4:64.18.0.0/20 ip4:74.125.148.0/22
ip4:74.125.244.0/22 ip4:207.126.144.0/20 include:_spf.google.com ~all
```

  - A couple of IP address ranges are allowed (CIDR notation)
  - Otherwise, google.com’s spf records must be checked
  - Otherwise soft fail; i.e. accept but mark as suspicious to trigger some other checks (beyond the scope of SPF)

SPF on the Receiver Side

- Receiver first gets the IP address of the sender (TCP connection)
- Locates SPF policies for the domains listed as HELO and MAIL FROM identities
- If the IP address is allowed for the domains as senders than SMTP continues with mail content transfer
    - Otherwise, stops

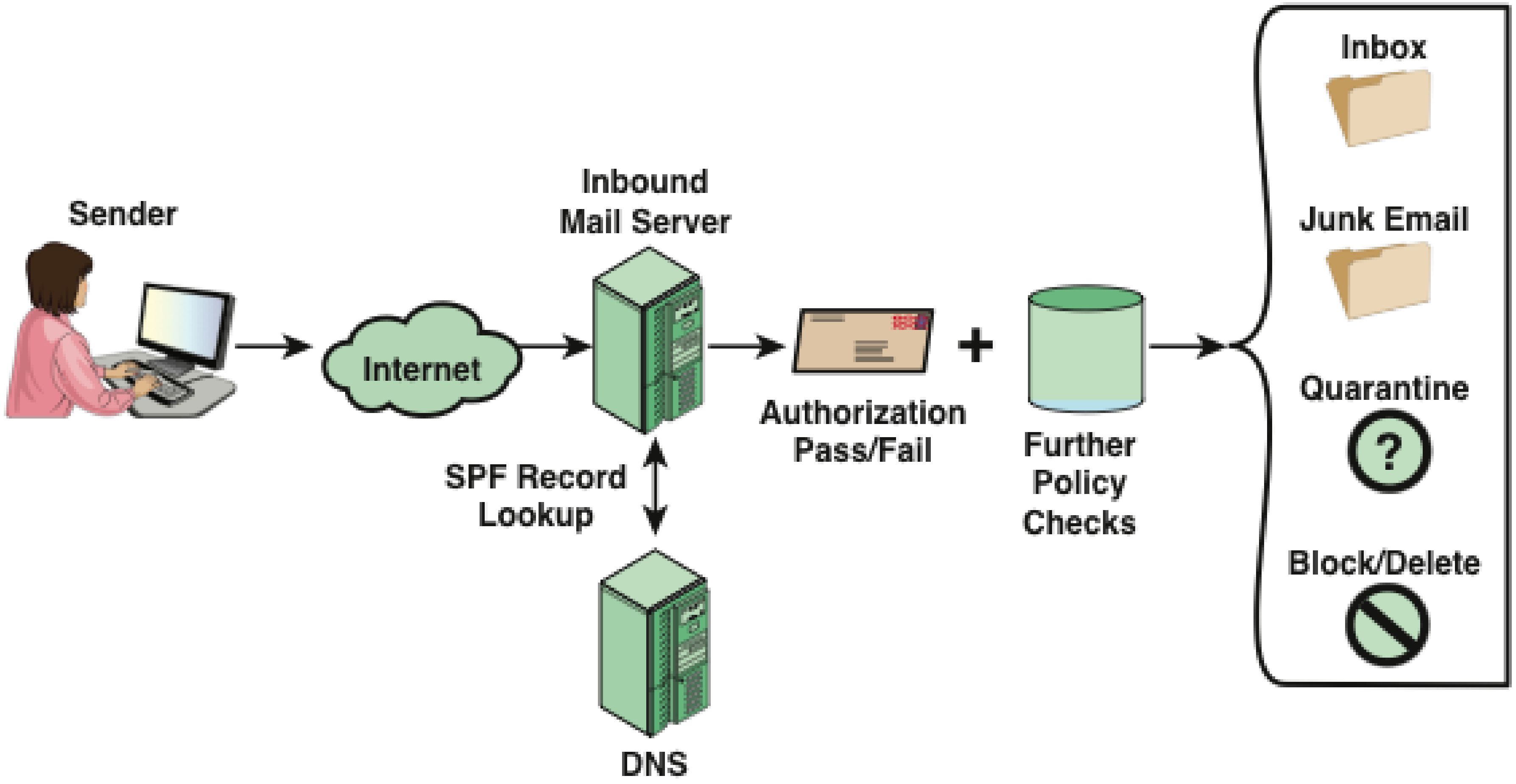


Figure 19.9 Sender Policy Framework Operation

Thank You!

Email Safeness Scorer

(EmailSafe)

EmailSafe

- Spam emails are unsolicited emails sent for either advertising the product and services or for phishing gullible users for sharing important and sensitive information.
  - Spams are undesired emails, sent in unsolicited bulk for advertising, phishing or inflicting malware on the recipient system.
- The EmailSafe is an application which would provide the safeness score of the emails based on its content,
  - low safeness score signifies the dubious, scam, phishing, malicious or spam emails
  - high score tells that the email is safe i.e. the email is free from any malicious intentions.

Solving Spam Problem

- Various algorithms have been devised for detecting spams.
- Some are successful and effective on the other hand others are highly biased and need regular training.
- We have created a multi layer neural network and trained it with a set of email data, and provided the scope for regular training with new examples.

Types of Email Phishing

Deceptive Phishing

- This is the most common type of phishing attack wherein a cybercriminal impersonates a known popular entity, domain or organization in the email and attempt to steal sensitive private information from the victim such as login, password, bank account detail, credit card detail, etc.
- This type of attack lacks sophistication as it does not have personalization and customization for the individuals.
- For an example, emails containing Phishing URL is disseminated in bulk to large users as a volume of mail is very high the cybercriminal would expect that many users will open the emails and visit the malicious URLs or open the infected attachments
- The email subject will be such that it might create urgency such as "Your account has been hacked, change your password immediately!", "Your bill is overdue-pay immediately of pay fine!" or other similar messages, once a user open such messages or visit the URLs the damage is done

Spear Phishing

- In this type of phishing emails contain an abundance of personalization information about the prospective victim.
    - The email might contain the name, company name, designation or his friends, colleagues and other social information of the recipient.
    - The proliferation of the company website, personal website and social media enables cybercriminals to get such details and assist them in forging a very convincing email.

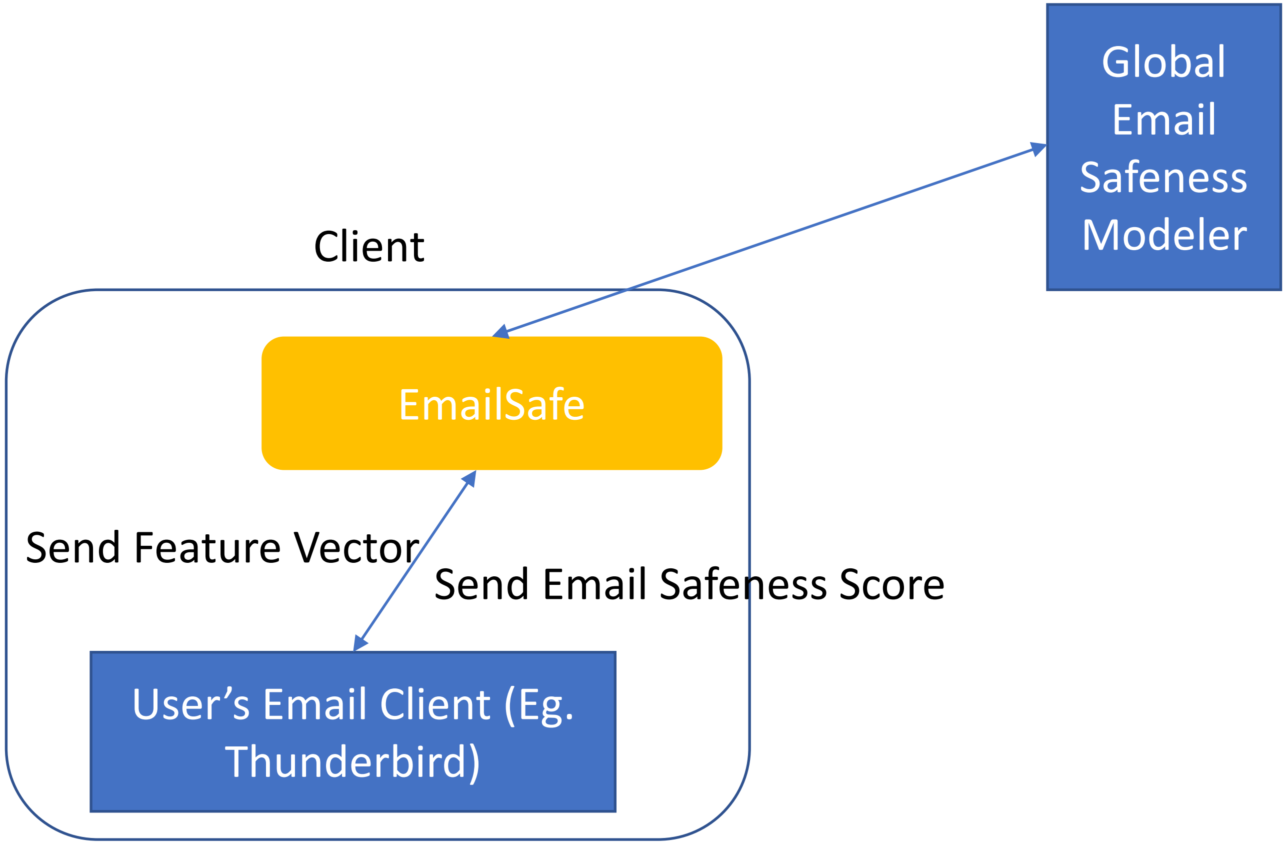
Whale Phishing

- This type of phishing targets business leaders such as CEO of top-level management employees to spear phish a "whale", here top-level executive such as CEO.
- The main aim of this type of phishing is to gather confidential information from the CEO and impersonate as CEO.
- This attack can render maximum damage to company financial prospects, market value, and reputation.

EmailSafe

- EmailSafe
  - A client side component that contains a NN model
  - Emails (in the form o feature vector) from Email client software are sent to ‘EmailSafe’ that decides on the safeness of the given email.
  - NN model in EmailSafe is updated regularly from a global Email Safeness Modeler
  - EmailSafe over time develops a customized safeness scoring system for the given user with the local and global knowledge;
    - Model is safe and within the control of the user;
    - Multiple ‘personas’ could be created – a custom NN model for each persona of the user
  - Privacy is preserved
    - Inputs are not taken from the EmailSafe by the global scorer

Block Diagram of Email Safeness Scorer



Client

EmailSafe

Global Email Safeness Modeler

Send Feature Vector

Send Email Safeness Score

User's Email Client (Eg. Thunderbird)

Email Safeness Scorer for Thunderbird

- Brief Description

- A plug-in for Mozilla Thunderbird(a popular email client) that would display the safeness score for the email received based on its content.
- The EmailSafe has following components.
  - EmailSafe contains a trained NN model with the email content safeness analysing capability.
  - Client part, a plug-in for Mozilla Thunderbird that would send the email content to the EmailSafe asynchronously for analysis and would display the safety score returned from the EmailSafe.
- Global Email Safeness Modeler
  - This component continuously enhances its intelligence by learning from new emails and patterns contributed to it in its SPAMBOX component by the users and builds a new model and sends it periodically to EmailSafe.

Deployment Methodologies

- EmailSafe can run in small memory footprint device (or) in the same machine as the client.
- The installer for Mozilla Thunderbird plug-in is already made available.
  - The plug-in can be made available for download at Mozilla Thunderbird extension store.

Potential societal impact of the research

- EmaiSafe application would classify the SPAM, SCAMS emails, Phishing emails and gives safeness scores to the users
    - EmailSafe could alert the user against exploitation from the malicious emails and potential financial loss as well as loss in reputation.

Publishing of work

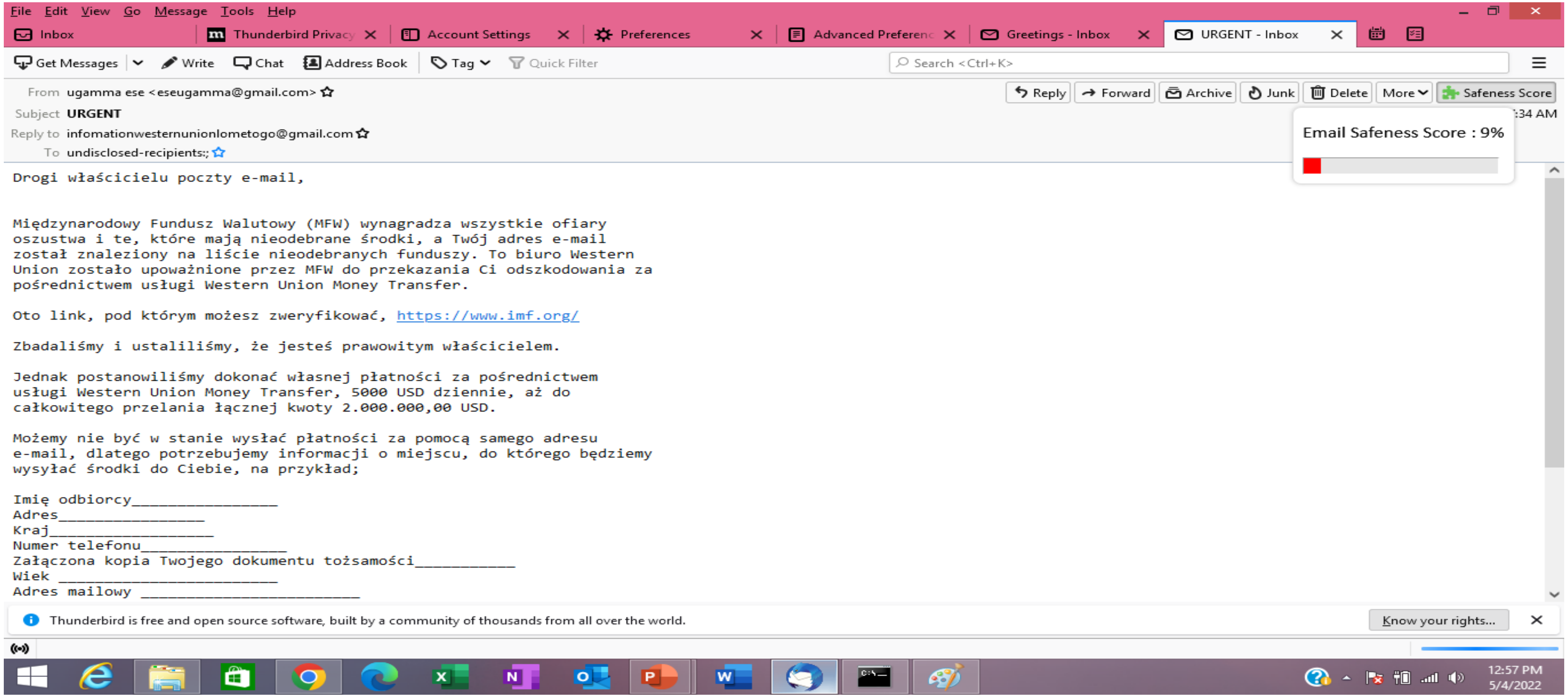
- A paper, “An Adaptive Neural Network for Email Spam Classification” published in IEEE-Fifteenth International Conference on Information Processing (ICINPRO), 2019 held at Capitol Hotel, Bengaluru, DOI: 10.1109/ICInPro47689.2019

Efficiency with respect to Functionality, Performance and Cost

- The proposed algorithm achieves a better performance without compromising in accuracy compared with existing heuristic approach.
- In an experiment involving same data, similar conditions, multiple methods for email safeness detection was implemented and tested and our method yielded 99.12 % of accuracy.

Strengths and limitations

- The proposed approach(ML approach) can well adapt to new email spam patterns and continuous learning at Email Safeness Modeler with personalized orientation in learning gives this a advantage over the rule based engine approach.
- The current EmailSafe Model is trained with 4000 email dataset (contains equal no. of spam and non spam) and feature vector is 1900 one dimensional matrix containing 0 and 1.
  - Better accuracy requires a large dataset and also new emerging spam email pattern.
- This approach relies on the content of the email and hence for mail having very few words or blank mail, the approach would resist to work.



From ugamma ese <eseugamma@gmail.com>
Subject **URGENT**
Reply to infomationwesternunionlometogo@gmail.com
To undisclosed-recipients:;

Email Safeness Score : 9%

Drogi właścicielu poczty e-mail,

Międzynarodowy Fundusz Walutowy (MFW) wynagradza wszystkie ofiary oszustwa i te, które mają nieodebrane środki, a Twój adres e-mail został znaleziony na liście nieodebranych funduszy. To biuro Western Union zostało upoważnione przez MFW do przekazania Ci odszkodowania za pośrednictwem usługi Western Union Money Transfer.

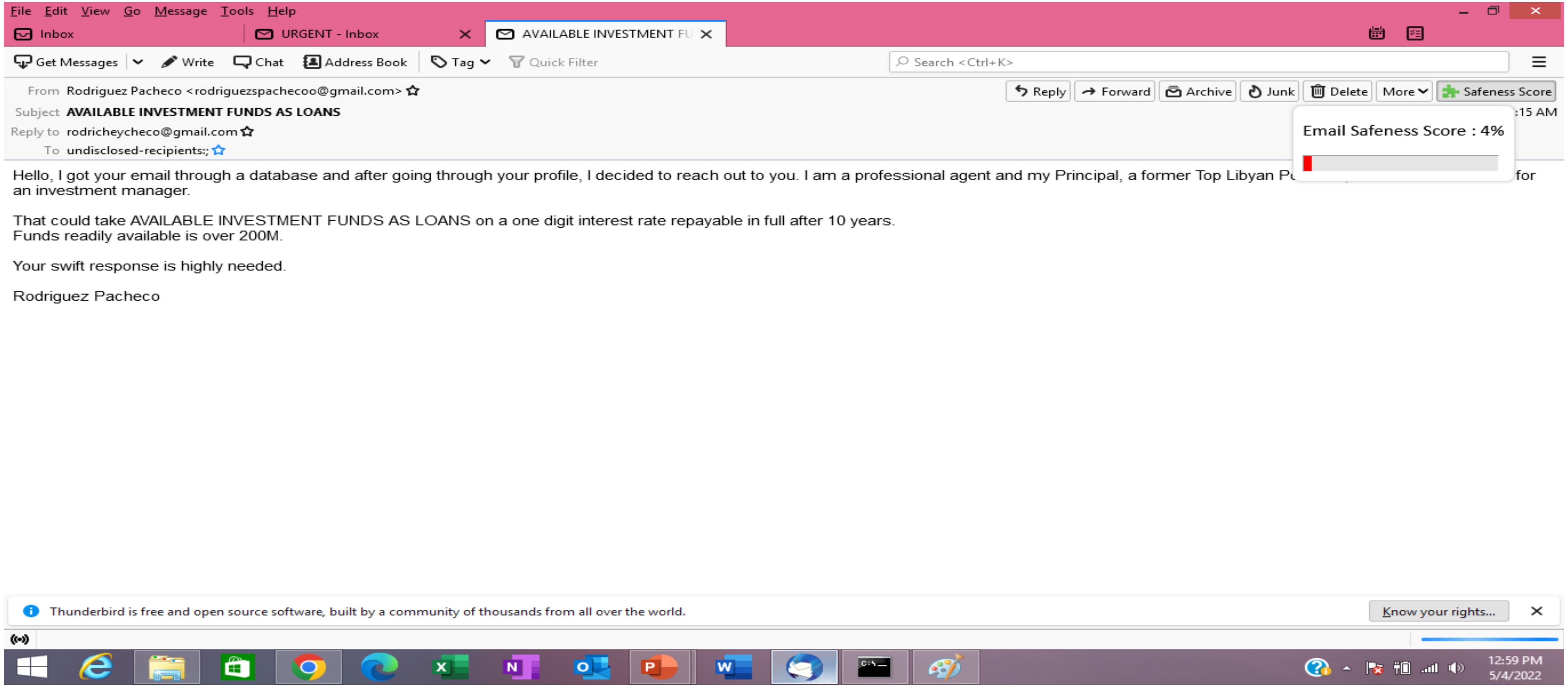
Oto link, pod którym możesz zweryfikować, https://www.imf.org/

Zbadaliśmy i ustaliliśmy, że jesteś prawowitym właścicielem.

Jednak postanowiliśmy dokonać własnej płatności za pośrednictwem usługi Western Union Money Transfer, 5000 USD dziennie, aż do całkowitego przelania łącznej kwoty 2.000.000,00 USD.

Możemy nie być w stanie wysłać płatności za pomocą samego adresu e-mail, dlatego potrzebujemy informacji o miejscu, do którego będziemy wysyłać środki do Ciebie, na przykład;

Imię odbiorcy________________
Adres________________
Kraj________________
Numer telefonu________________
Załączona kopia Twojego dokumentu tożsamości___________
Wiek ________________________
Adres mailowy ________________________



From Rodriguez Pacheco <rodriguezspachecoo@gmail.com>

Subject **AVAILABLE INVESTMENT FUNDS AS LOANS**

Reply to rodricheycheco@gmail.com

To undisclosed-recipients:;

:15 AM

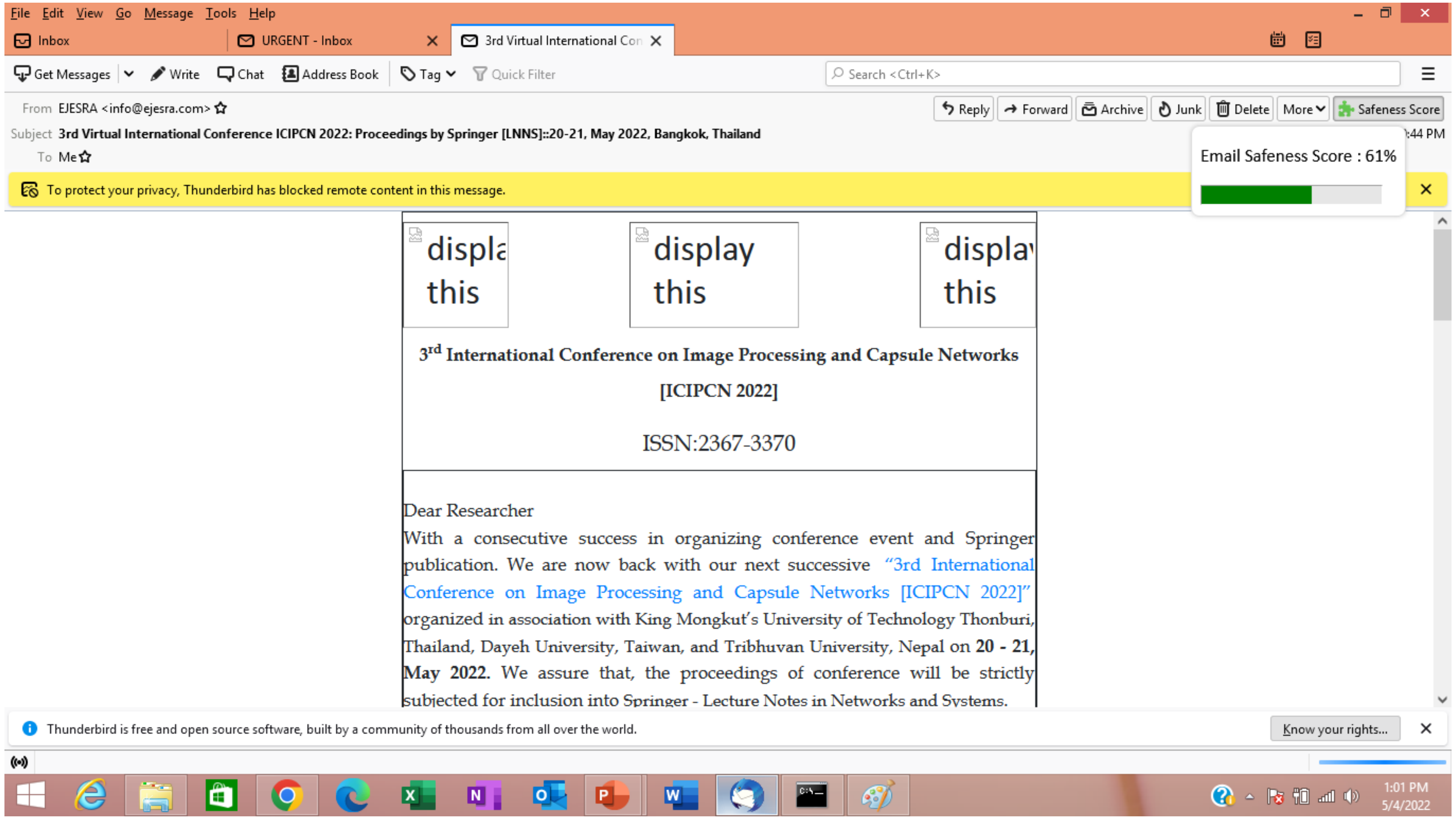
Email Safeness Score : 4%

Hello, I got your email through a database and after going through your profile, I decided to reach out to you. I am a professional agent and my Principal, a former Top Libyan Po for an investment manager.

That could take AVAILABLE INVESTMENT FUNDS AS LOANS on a one digit interest rate repayable in full after 10 years.
Funds readily available is over 200M.

Your swift response is highly needed.

Rodriguez Pacheco



From EJESRA <info@ejesra.com>

Subject 3rd Virtual International Conference ICIPCN 2022: Proceedings by Springer [LNNS]::20-21, May 2022, Bangkok, Thailand

To Me

Email Safeness Score : 61%

To protect your privacy, Thunderbird has blocked remote content in this message.

3^{rd} International Conference on Image Processing and Capsule Networks [ICIPCN 2022]

ISSN:2367-3370

Dear Researcher

With a consecutive success in organizing conference event and Springer publication. We are now back with our next successive "3rd International Conference on Image Processing and Capsule Networks [ICIPCN 2022]" organized in association with King Mongkut's University of Technology Thonburi, Thailand, Dayeh University, Taiwan, and Tribhuvan University, Nepal on **20 - 21, May 2022.** We assure that, the proceedings of conference will be strictly subjected for inclusion into Springer - Lecture Notes in Networks and Systems.

Thunderbird is free and open source software, built by a community of thousands from all over the world.

Spam Email Classification Using Adaptive Neural Network

Pre-processing of emails

- Lower-casing : The entire email is converted into lower case
- Stripping HTML : All HTML tags are removed from the emails.
- Normalizing URLs: All URLs are replaced with the text "httpaddr".
- Normalizing Email Addresses: All email addresses are replaced with the text "emailaddr".
- Normalizing Numbers: All numbers are replaced with the text "number".
- Normalizing Dollars: All dollar signs ($) are replaced with the text "dollar"
- Word Stemming: Words are reduced to their stemmed form. For example, "discount", "discounts", "discounted" and "discounting" are all replaced with "discount".
- Removal of non-words: Non-words and punctuation have been removed. All white spaces (tabs, newlines, spaces) have all been trimmed to a single space character.

Feature Selection and Extraction

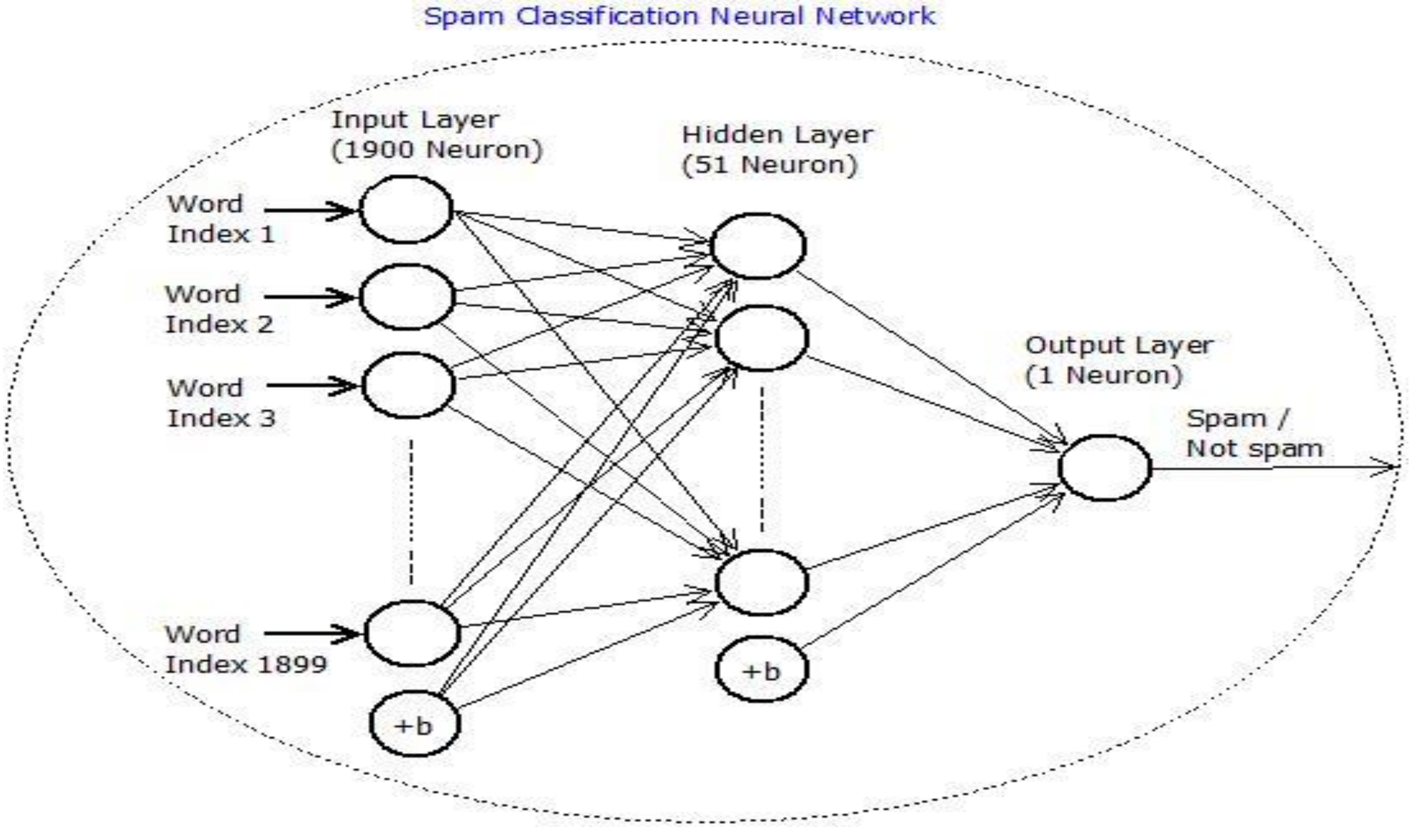
- Set of most frequent words from set of training emails are selected as features
  - The frequent words can be considered as vocabulary list.
- Leave those words which are occurring in very few emails of our training set,
  - might cause the model to overfit for the given training set.
- The vocabulary list of words should be selected by choosing the words which occurs at least 100 times in the spam corpus.
- The vocabulary list selected contains 1899 words
  - In practice, a vocabulary list size should be 10,000 to 50,000 for better accuracy. These 1899 words of the vocabulary list are the features for the classifier.

Dataset for Training

- For training, the dataset is based on a subset of the SpamAssassin Public Corpus.3 [12]. The file 'spam_features.mat' contains email data set for the training and is based on a subset of the SpamAssassin Public Corpus.3 .
- The 'spam_features.mat' is a matrix of 4000 spam emails data with 1899 features.

```
mat =loadmat('spam_features.mat',)

X = mat['X’] # Features Matrix of spam data

y = mat['y’] # Class label matrix of spam data

/* “X_train” is a dataset for training and “y_train” is its corresponding class labels
and similarly “X_test” is a dataset for testing and  “y_test” its corresponding class
labels

*/

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

Three Layer Neural Network for testing the algorithm

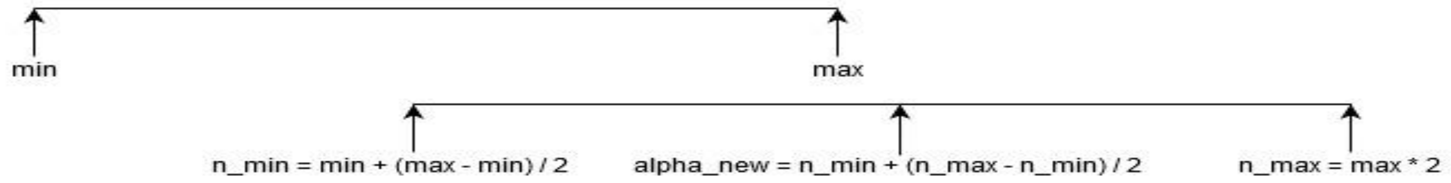


Backpropagation Algorithm for Neural Network

```
function BPNN (theta1, theta2, alpha, iterations) {
for it in range (iterations) {
grad, error = nnCostFunction(theta1,theta2,input_layer_size,hidden_layer_size,num_labels,X_train,y_train,lamda);
bpnn_error += error;
Theta1_gradient =grad[0][0];
Theta2_gradient = grad[1][0];
theta1 = theta1 - alpha*Theta1_gradient;
theta2 = theta2 - alpha*Theta2_gradient;
}
THETA = [[theta1],[theta2]];
avg_error = bpnn_error / iterations;
return THETA , avg_error;
}
```

Basic Idea of Adaptive Neural Network Algorithm

- Start with a low value of alpha (learning rate) at beginning and calculate the initial average cost/average error for the backpropagation algorithm(BPNN) for some small fixed number of iterations.
- Call the adaptive neural network function on initial values of min=0 and max=alpha and alpha (learning rate) .


min
max
n_min = min + (max - min) / 2
alpha_new = n_min + (n_max - n_min) / 2
n_max = max * 2

- The adaptive neural network function call BPNN which would return the average error for a small fixed number of iterations for the new values mentioned in above diagram.
- If average error decreases then recursively call the adaptive neural network function with new values.
- Else complete the remaining iterations of backpropagation algorithm with the old values.

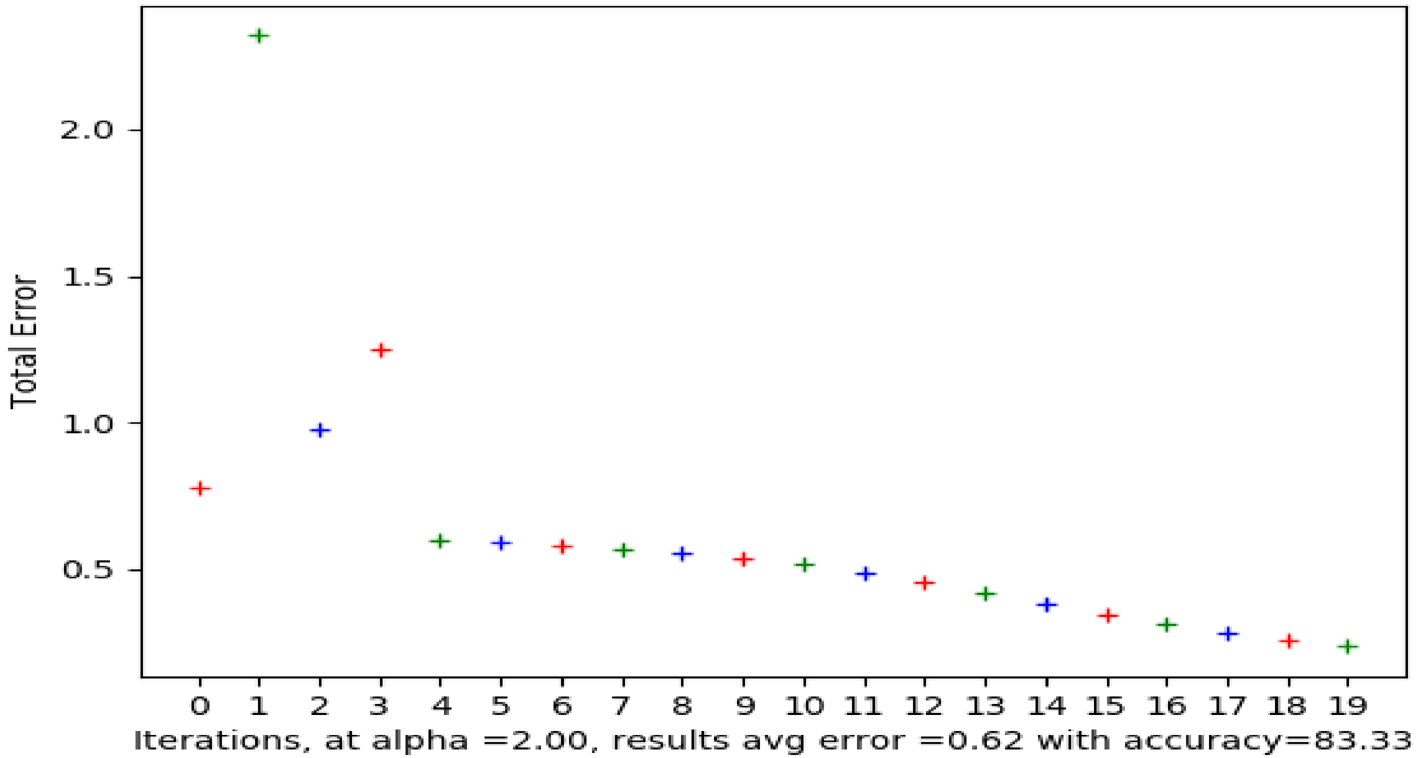
An Adaptive Neural Network Algorithm

```
function Adaptive_Neural_Network(THETA, min, max, initial_error, alpha, total_iterations){
    ac_score = accuracy_test(THETA);
    if (ac_score > 99.5) return THETA, alpha;
    n_min=min+(max-min)/2;
    n_max = max *2;
    alpha_new=n_min+(n_max-n_min)/2;
    THETA, ErrorTotal=BPNN(THETA, alpha_new ,test_iterations);
    remaining_iterations=total_iterations-test_iterations;
    if(ErrorTotal-initial_error<0)
        Adaptive_Neural_Network(THETA, n_min, n_max, ErrorTotal, alpha_new, remaining_iterations );
    else {
        THETA,ErrorTotal=BPNN(THETA, alpha ,remaining_iterations)
        return THETA, alpha
} }
```

Execution of algorithm

Execution of adaptive neural network algorithm for alpha=2.0 and total iterations=200 and test_iterations=20

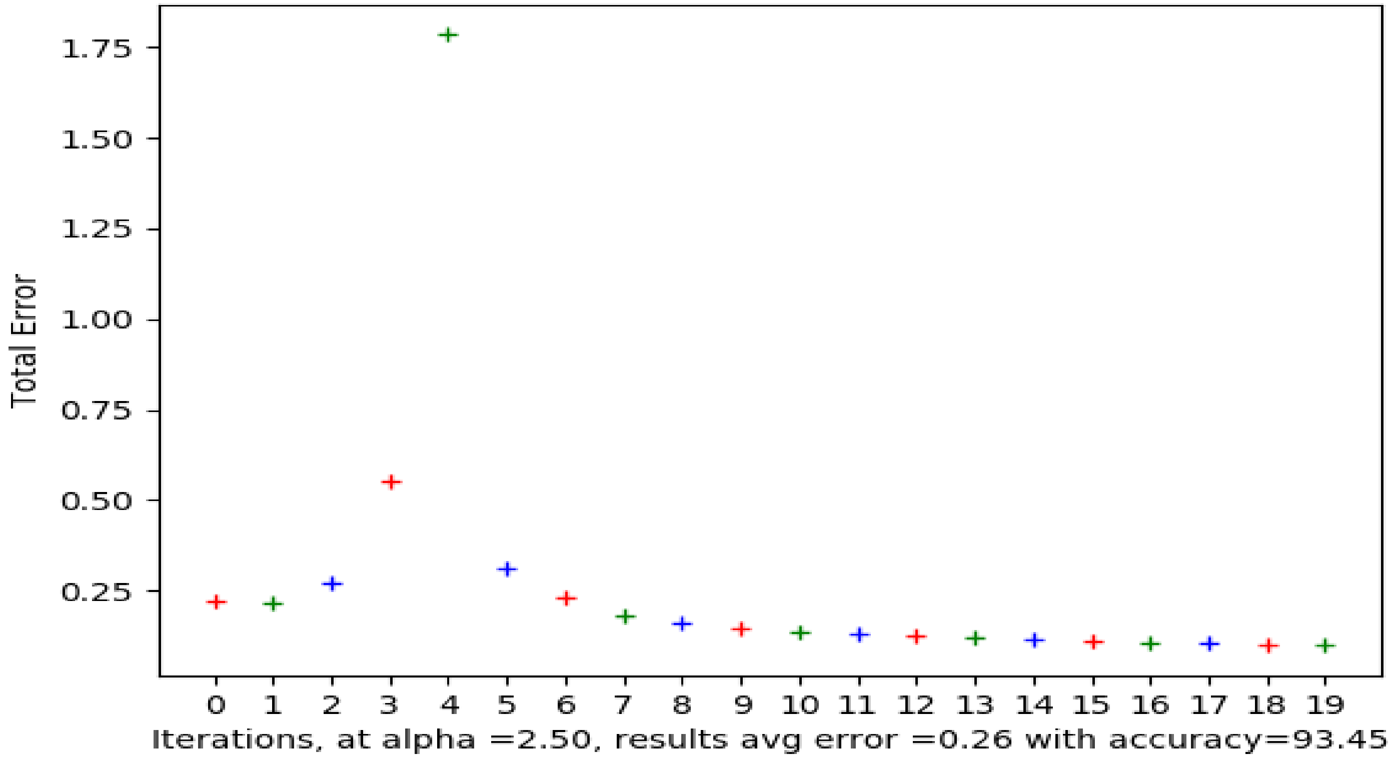
Results Analysis



First Call of the BPNN() function

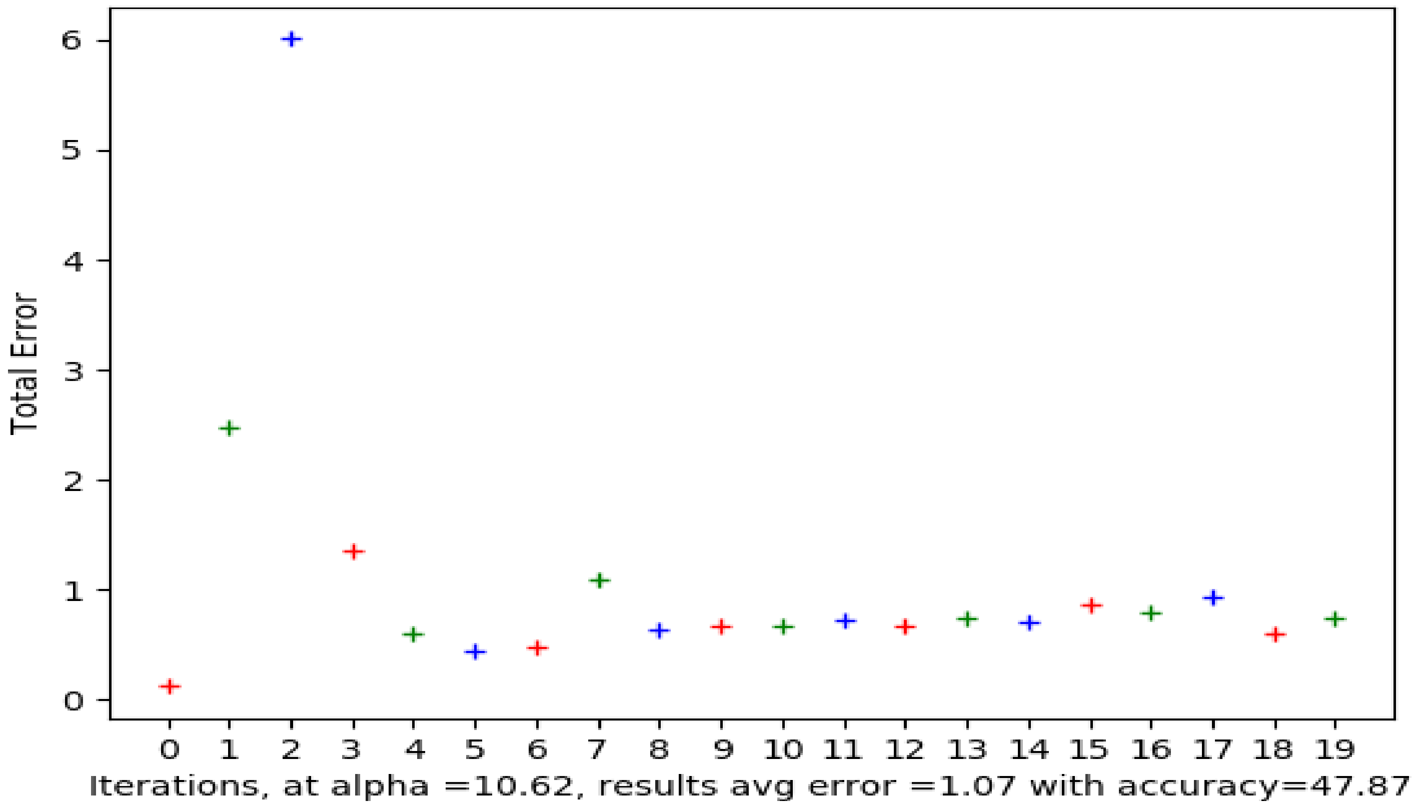
Results Analysis

- First Ca



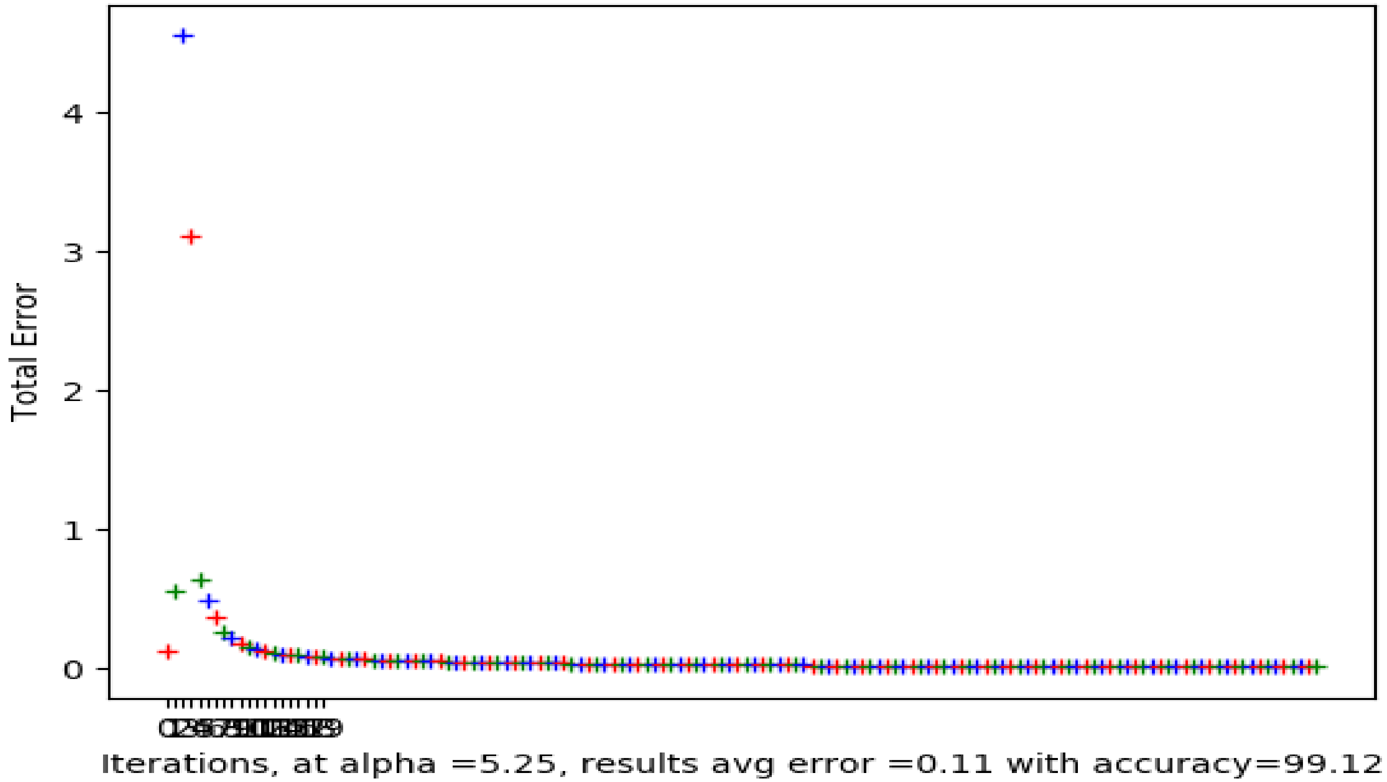
Second call of the BPNN() function

Results Analysis



Fourth call of the BPNN() function

Results Analvsis



Finally BPNN() the algorithm continues for 120 remaining iterations

Comparison

Classification Algorithms	Accuracy
Adaptive Neural Network (proposed in the paper, with 200 iterations and initial learning rate=2)	99.12 %
Logistic Regression	98.25%
SVM	96 %
MLP Classifier(with 200 iterations and learning rate=2)	98 %

Conclusion

- The proposed algorithm adaptively changes the best-suited learning rate during its recursive calls and returns the neural network optimized weights which results in high accuracy.
- The proposed Adaptive Neural Network based on adaptive learning rate performs better and have fast convergence and less prone to overfitting problem.

Thank You