PrivDPI: Privacy-Preserving Encrypted Traffic Inspection with Reusable Obfuscated Rules

Jianting Ning, Geong Sen Poh, Jia-Ch'ng Loh,
Jason Chia, Ee-Chien Chang

Outline

- Introduction
- Motivation
- BlindBox by Sherry et al. in Sigcomm 2015
- PrivDPI
- Performance Evaluation
- Prototype (Screenshots)
- Conclusion

NUS
National University
of Singapore

Introduction

Introduction

- **Popularity of TLS**
  - More than 80% web traffic would be encrypted by TLS in 2019. [1]
  - Major browser implementations like Firefox, Chrome only support HTTP/2 over TLS. [2]

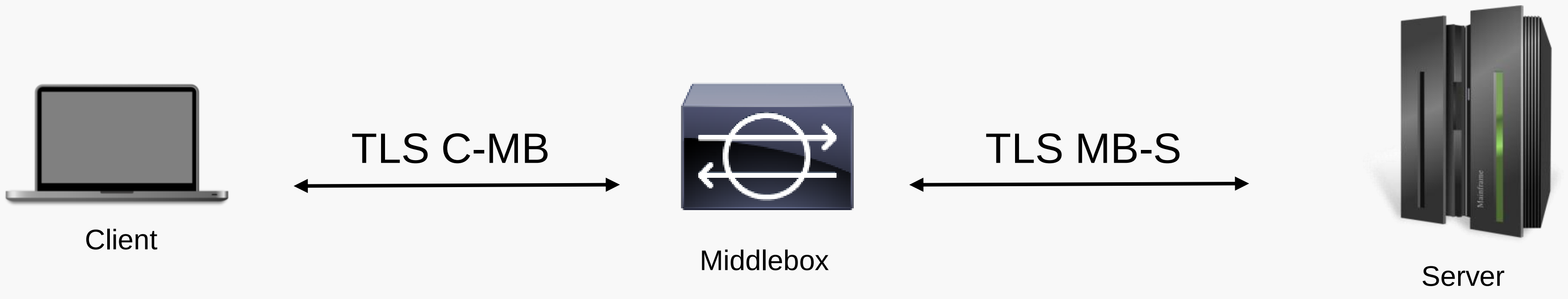
- **Easy cover for attackers**
  - Firewall, IDS, IPS, and any Middlebox (MB) in an organization might not function well at this point.
  - Almost 50% of the cyber attacks use encryption as cover to sneak into organization network. [3]

[1]https://www.cisco.com/c/dam/en/us/solutions/collateral/enterprise-networks/enterprise-network-security/nb-09-encrytd-traf-anlytcs-wp-cte-en.pdf
[2]https://en.wikipedia.org/wiki/HTTP/2#Encryption
[3]https://www.computerweekly.com/news/450303346/Encryption-hiding-malware-in-half-of-cyber-attacks

Solution: Man in the Middle (MitM)

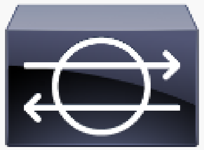
How can MB inspect ?

- Current industry practice: MitM
- Performs man-in-the-middle to decrypt the traffic to perform DPI.
- Client/server is aware to be inspected.



Motivation

Motivation



- **MitM approach is working fine in on-premises MB**
  - Data is still within an organization

- **Main issue: Cloud-based (Third party) MB**

  Security:
  - Endpoints need to trust the middlebox (MB). Violates end-to-end security guarantee of TLS.
  - Middlebox may weaken the security of TLS by using obsolete security parameters.
  - The various security issues prompted the US-Cert to issue an alert (TA17-075A) on interception of encrypted traffic [4].

[4]https://www.us-cert.gov/ncas/alerts/TA17-075A

BlindBox

BlindBox: Deep Packet Inspection over Encrypted Traffic

- Proposed by Sherry et al. (UC Berkeley), *Sigcomm 2015*
- Match encrypted tokens of network traffic with encrypted rules

- **Advantages :**
    - *Privacy-preserving*
        - MB inspects over client's encrypted traffics without decryption
        - Client and server do not learn the rules
- **Use cases:**
    - Data exfiltration
    - Parental filtering
    - Ensure privacy of employees' encrypted traffic
- **Issues:**
    - **Every** session requires **setup** of encrypted rules which requires the communication overhead: **97s** and **50GB** for 3000 rules

Justine Sherry, Chang Lan, Raluca Ada Popa, Sylvia Ratnasamy: BlindBox: Deep Packet Inspection over Encrypted Traffic. SIGCOMM 2015: 213-226

BlindBox: Deep Packet Inspection over Encrypted Traffic

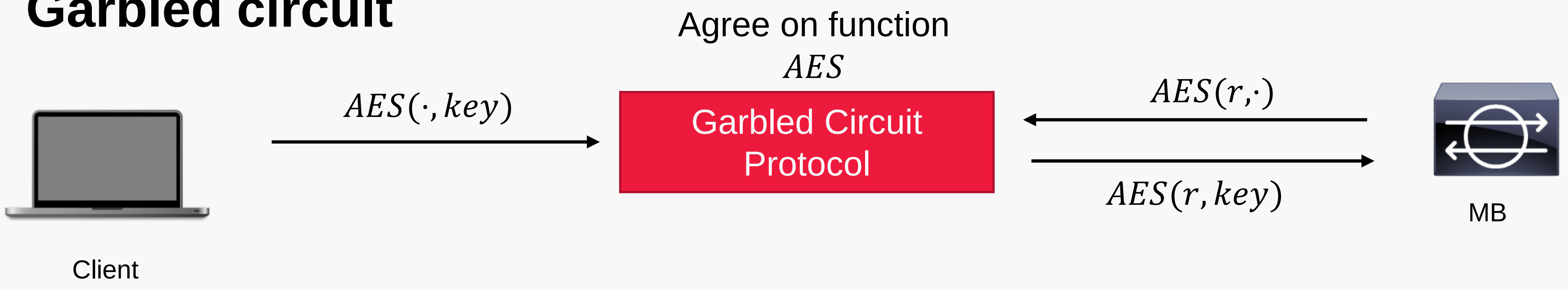
Two phases:

- **Setup:**
    - To generate encrypted rules (encrypted using session key).
    - Requirement: MB does not learn the session key and client/server does not learn the rules
    - Using circuit garbling hence it is compute intensive.

- **Token encryption & detection:**
    - Client/server tokenizes and encrypts the payload.
    - MB performs matching based on the encrypted tokens and the encrypted rules.

BlindBox: Deep Packet Inspection over Encrypted Traffic

- **Garbled circuit**



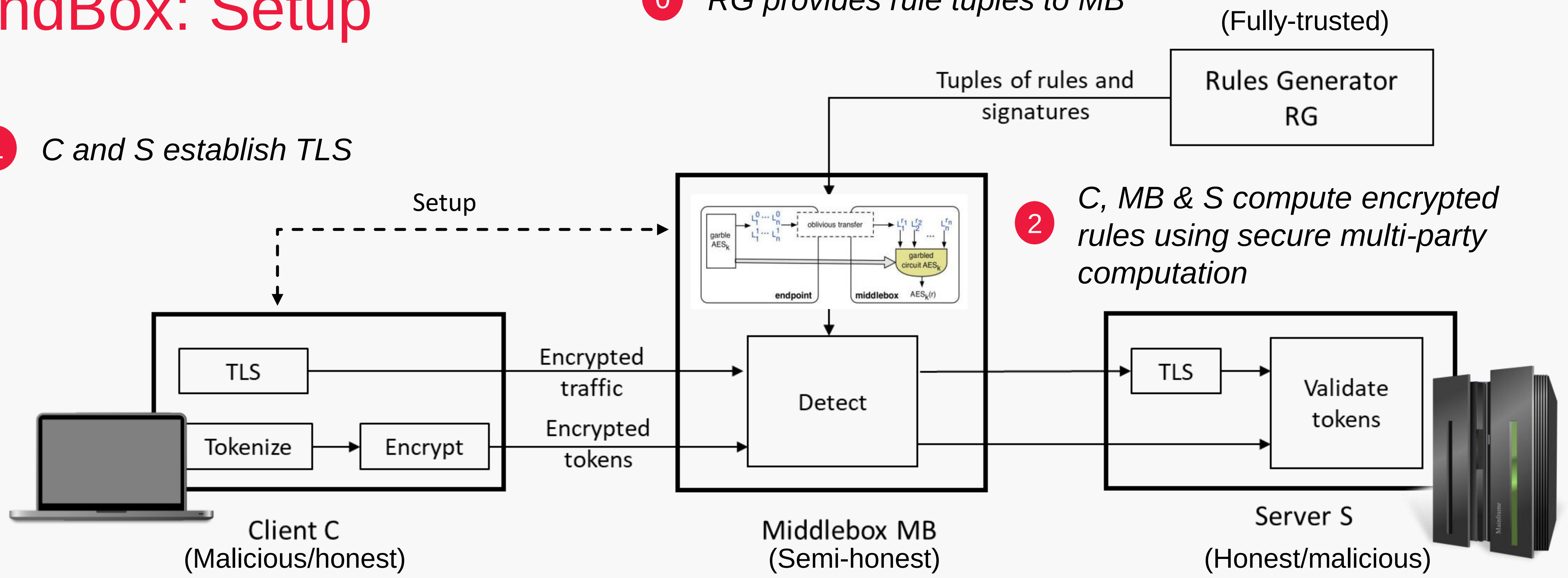
Justine Sherry, Chang Lan, Raluca Ada Popa, Sylvia Ratnasamy: BlindBox: Deep Packet Inspection over Encrypted Traffic. SIGCOMM 2015: 213-226

BlindBox: Setup

0 *RG provides rule tuples to MB*

1 *C and S establish TLS*

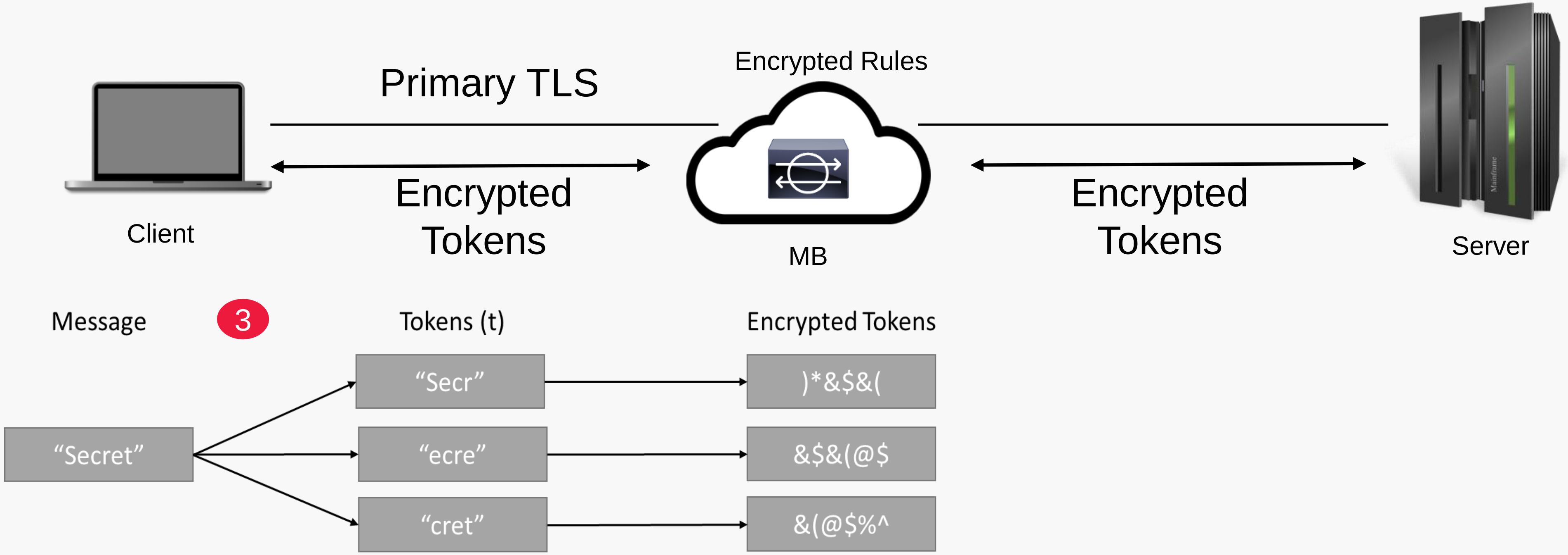
2 *C, MB & S compute encrypted rules using secure multi-party computation*



1. Must be performed for every session.
2. The operations are carried out after the session key has been established under the TLS handshake protocol.

Justine Sherry, Chang Lan, Raluca Ada Popa, Sylvia Ratnasamy: BlindBox: Deep Packet Inspection over Encrypted Traffic. SIGCOMM 2015: 213-226

BlindBox: Encrypted traffic inspection

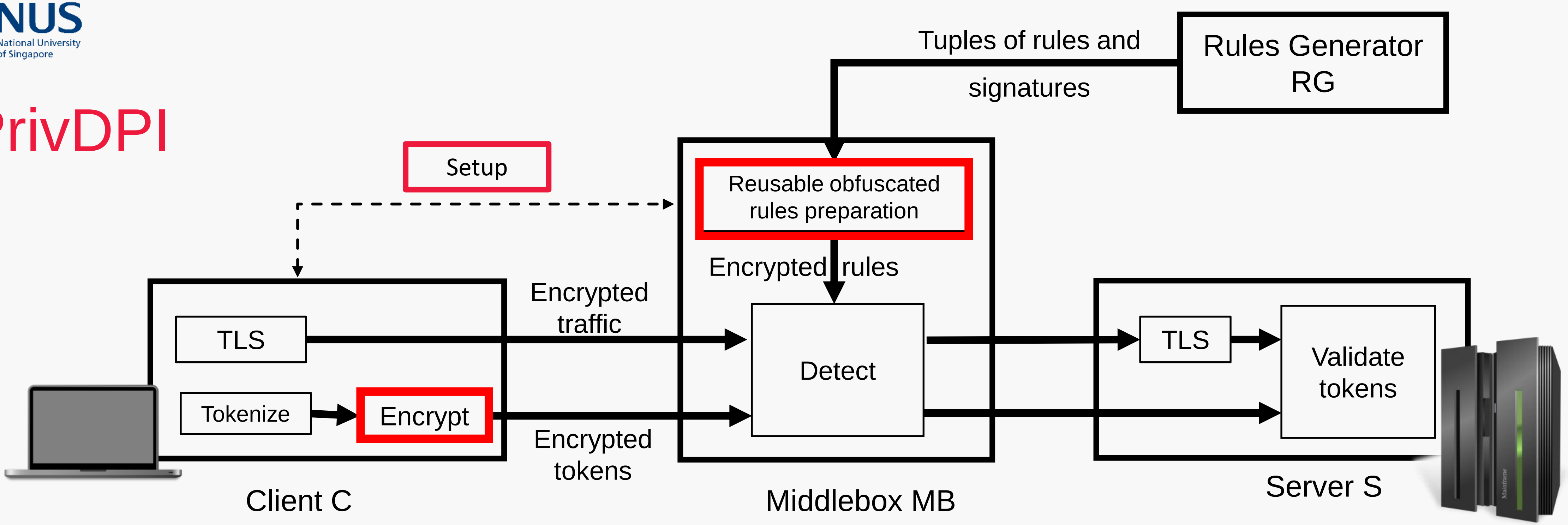


1. After setup, the client tokenizes (windows-based) and encrypts the payload using a key derived from the session key.
2. MB inspects encrypted tokens, and only tokens that match rules are revealed.

Justine Sherry, Chang Lan, Raluca Ada Popa, Sylvia Ratnasamy: BlindBox: Deep Packet Inspection over Encrypted Traffic. SIGCOMM 2015: 213-226

PrivDPI

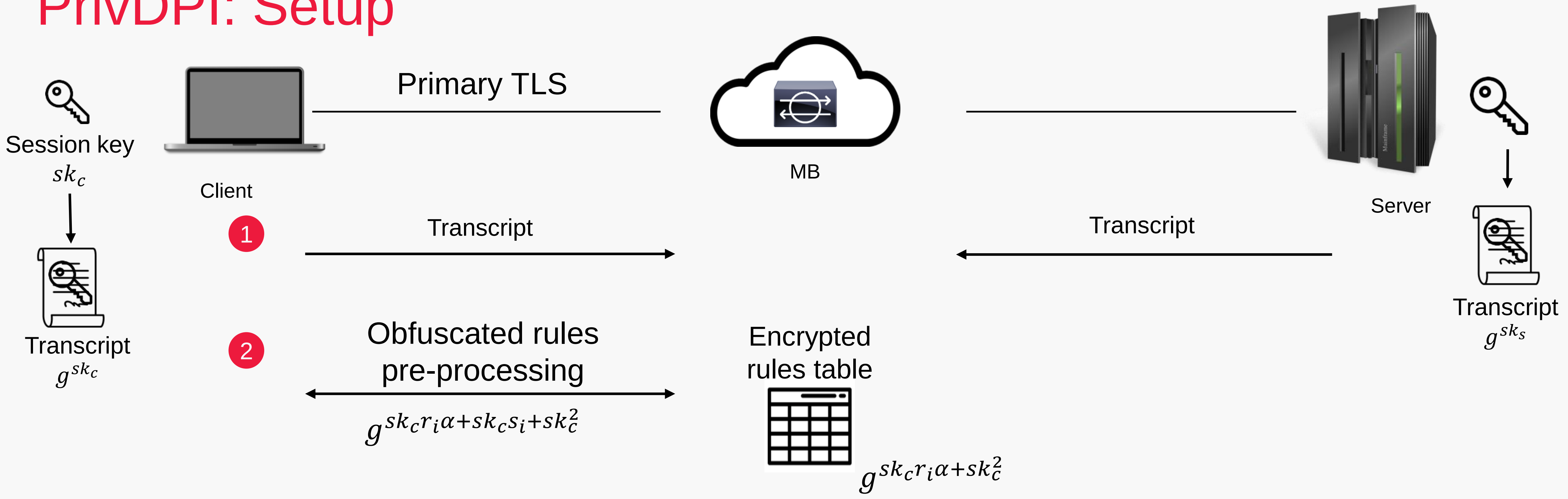
PrivDPI



1. Same setting like BlindBox but different approaches on **encrypting** the rules and tokens. New technique: **obfuscated rule encryption**.
2. **Session Reusable**: Does not need to perform the setup operation for every session.
3. **Reusable token encryption:** Reuse session token generated in previous sessions.

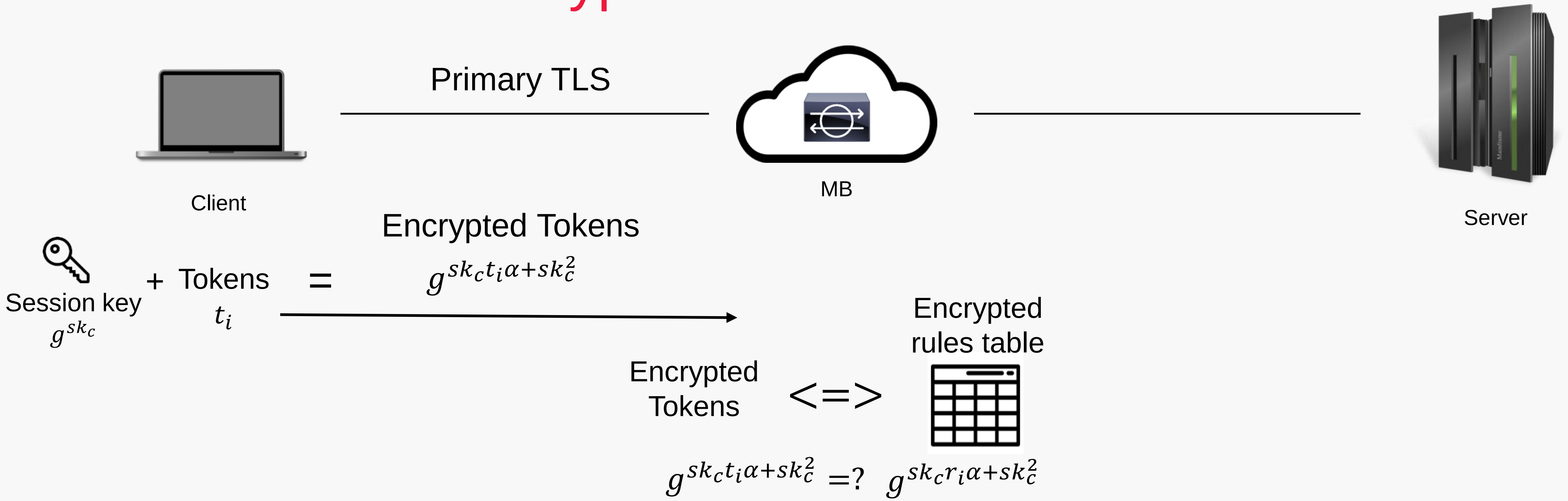
Computation is much more efficient and the bandwidth required is very low as compared to BlindBox.

PrivDPI: Setup



1. Client and server both forward the session token to MB:
   - MB can perform validation using both session tokens.
   - To prevent client being malicious, e.g. client uses random key in 2nd phase.
2. MB performs pre-processing with client to generate reusable encrypted rules table.
   - MB can use this table to detect encrypted tokens later.

PrivDPI: Token Encryption & Detection



MB checks if any encrypted token matches in the encrypted rules table.

PrivDPI: Session Reusable

Client

Primary TLS

MB

Server

sk_c (computed in first session)
sk_{c2} = $hash(k_{TLS}) \in Z_r$ (**new session**)

1

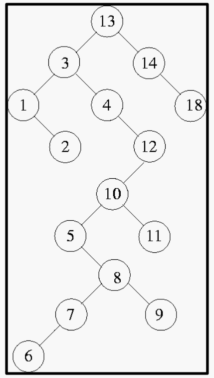
$pk_{c2} = g^{sk_{c2}}$,
$salt_{c2}$

Check:
$pk_{c2} =? \, pk_{s2}$
$salt_{c2} =? \, salt_{s2}$

$pk_{s2} = g^{sk_{s2}}$,
$salt_{s2}$

2

MB re-computes for every rule r_i:
$g^{sk_c r_i \alpha + sk_c^2 + sk_{c2}}$



Performance Evaluation

Comparison Table (Setup Time in second)

- Computer Specifications:
  - 6 Core - PC
  - Intel(R) Core(TM) i7-8750H CPU @ 2.20Ghz

- For 3,000 rules:
  - Blindbox takes around 3 minutes.
  - PrivDPI requires only 570 ms.

No. of Rules (8 bytes)	Setup Time	
	Blindbox	PrivDPI
300	17.8501	0.1041
600	35.7093	0.1107
900	52.6958	0.1669
1200	70.1994	0.2176
1500	90.7262	0.2717
1800	112.0800	0.3413
2100	132.9253	0.3978
2400	144.6417	0.4756
2700	165.1693	0.5253
3000	**183.8260**	**0.5683**

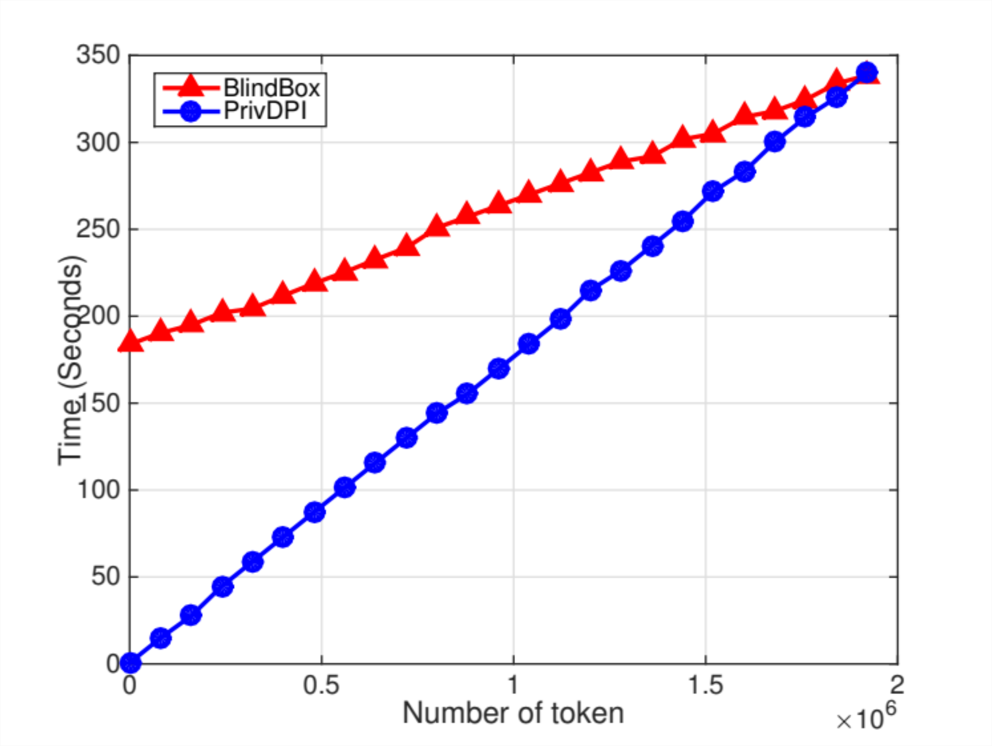
Comparison Table (Bandwidth required for each rule)

- In order to complete the setup phase, communications between Client (Server) and MB bandwidth is required.

- For 3,000 rules during the setup:
  - Blindbox takes **50.1 GB**.
  - PrivDPI takes only 324 KB.

Bandwidth required during Setup (for each rule)	
BlindBox	**PrivDPI**
16.7 MB	**108 B**

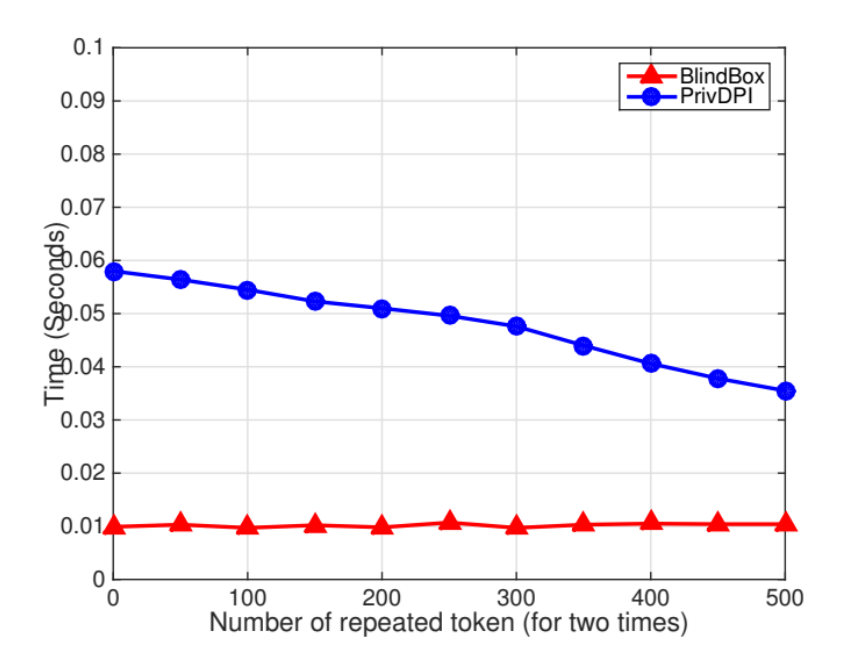
Tradeoff: Setup Time against Token Encryption Time

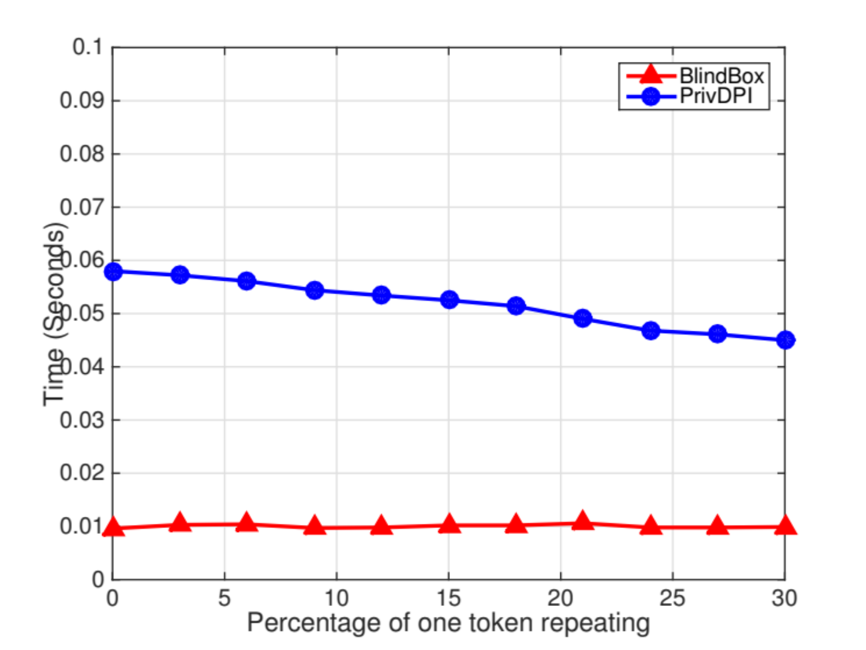


- Token encryption is traded-off for the setup time.
- Blindbox is more efficient after sending 3.6 million encrypted tokens.
- PrivDPI is more suitable for short-flowed communications due to the savings in the setup.
- E.g. sending e-mail or surfing bank details.

Reusable Token Encryption

- Optimization to increase the performance of token encryption.
- Assuming tokens reappear in a session or across different sessions.
- E.g. bloggers, documents, or webpages that are surfed frequently.

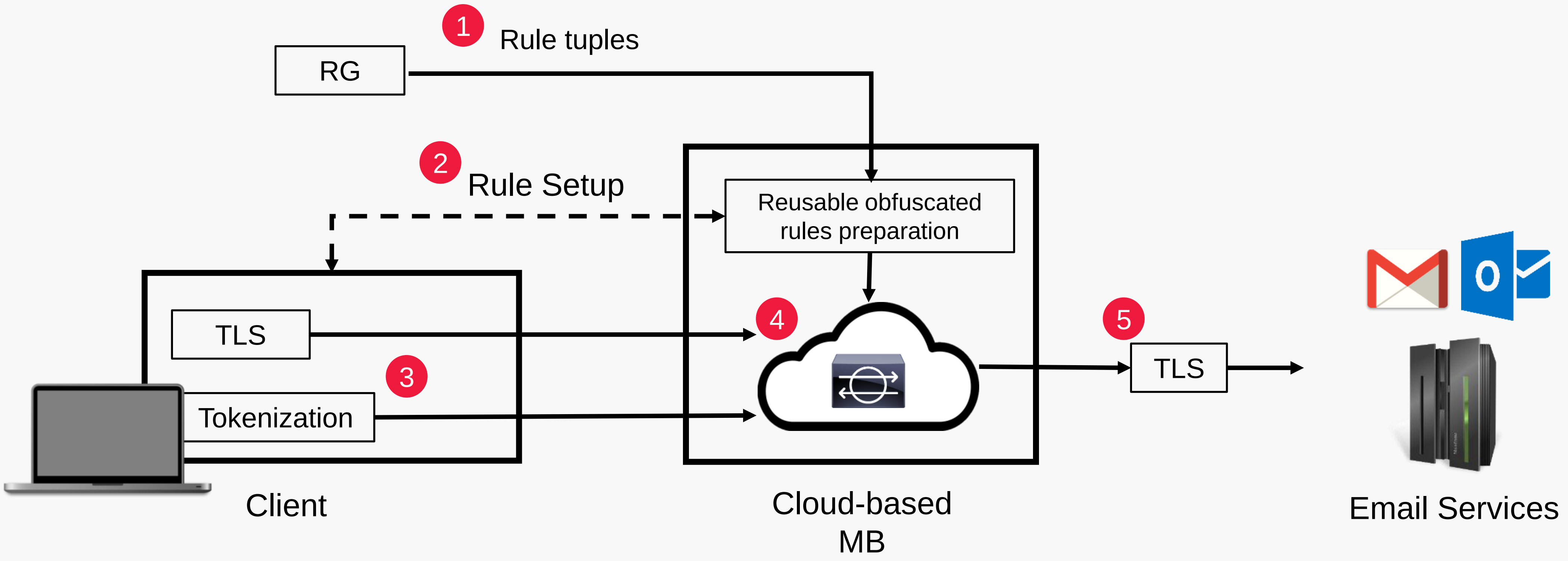




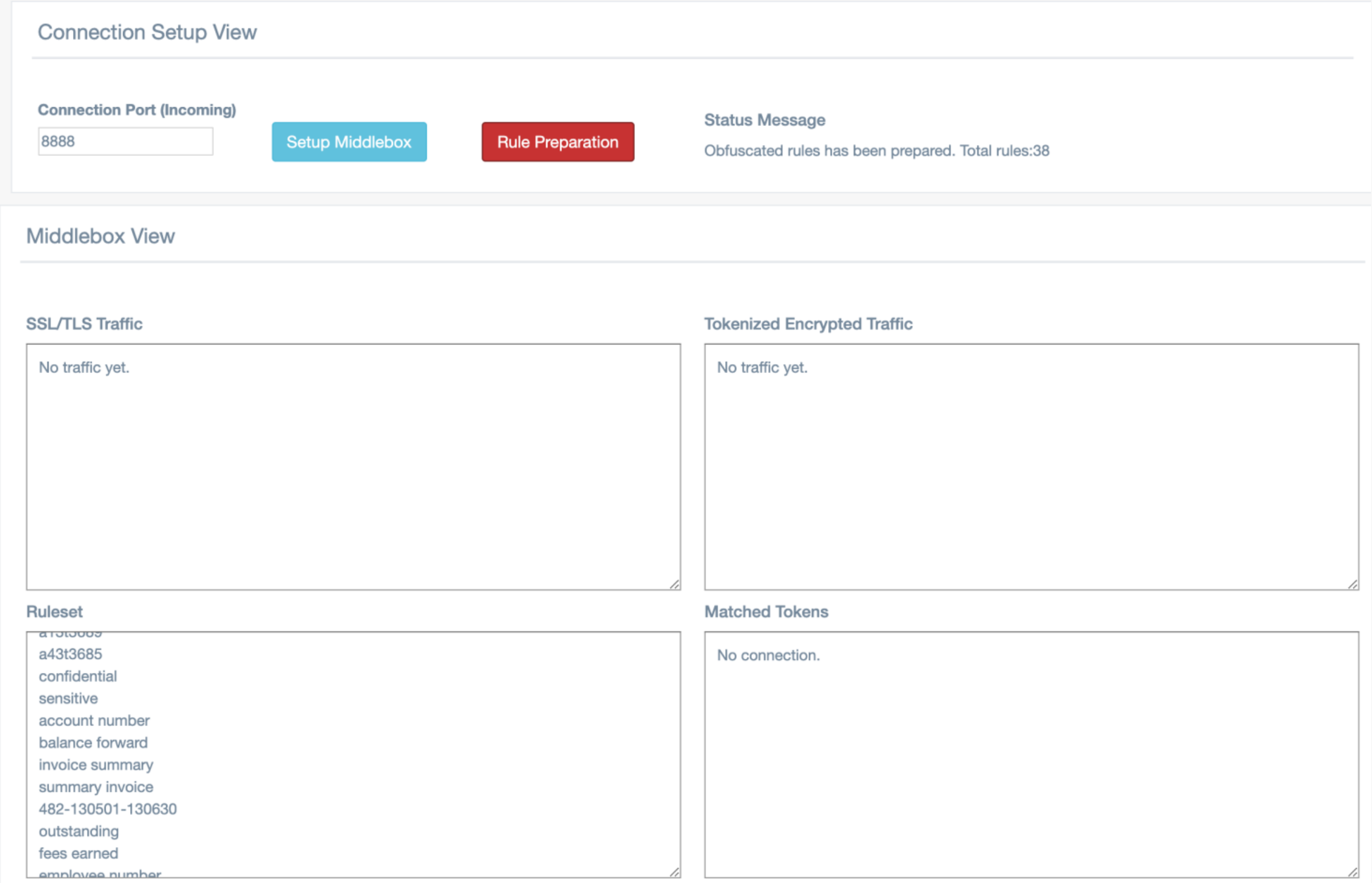
Prototype (Screenshots)

Privacy-preserving Secure Email Gateway

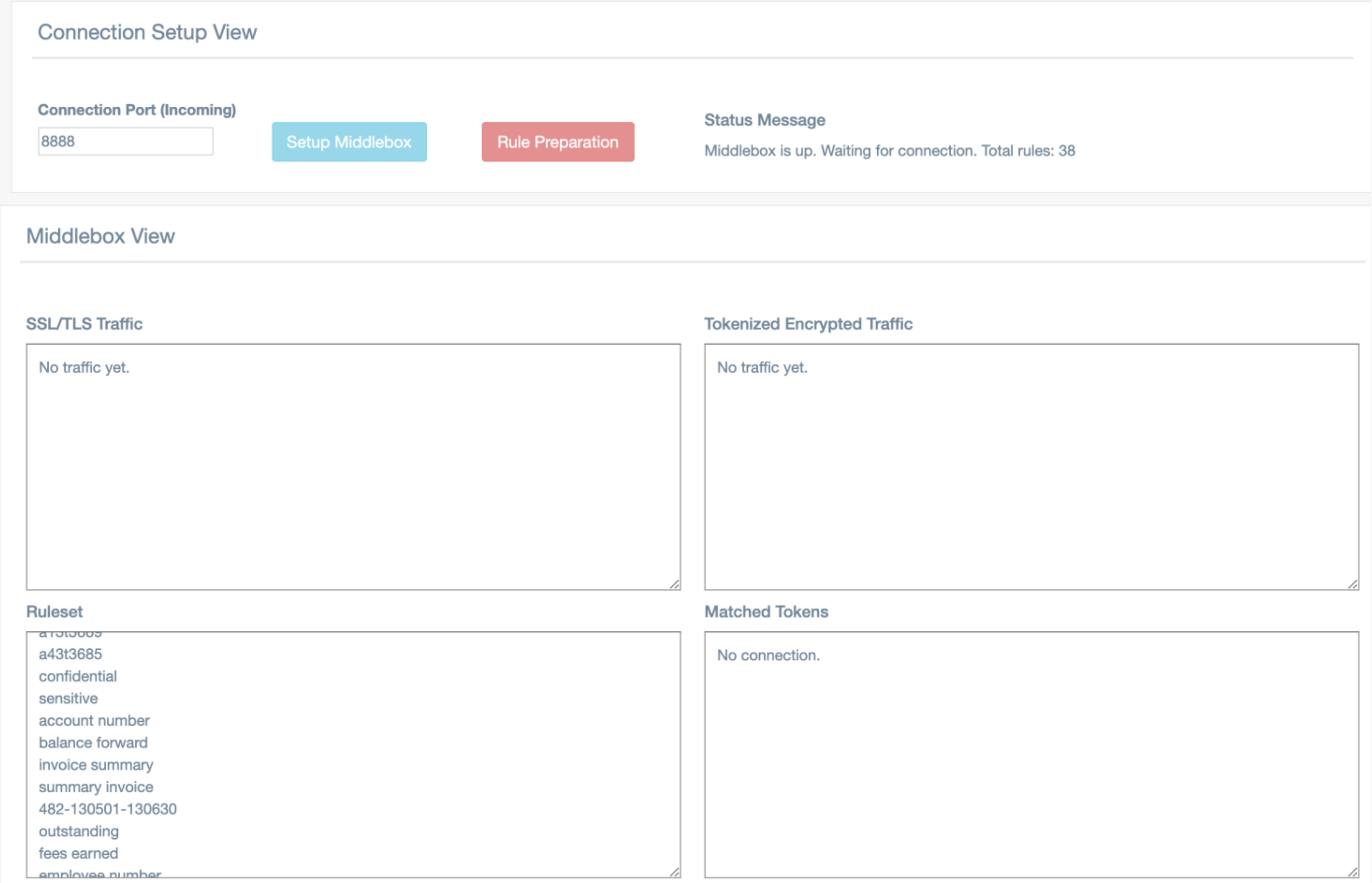
- Preserving privacy of employees
- Scan outbound emails and attachments
- Prevent sensitive data to go out of an organization



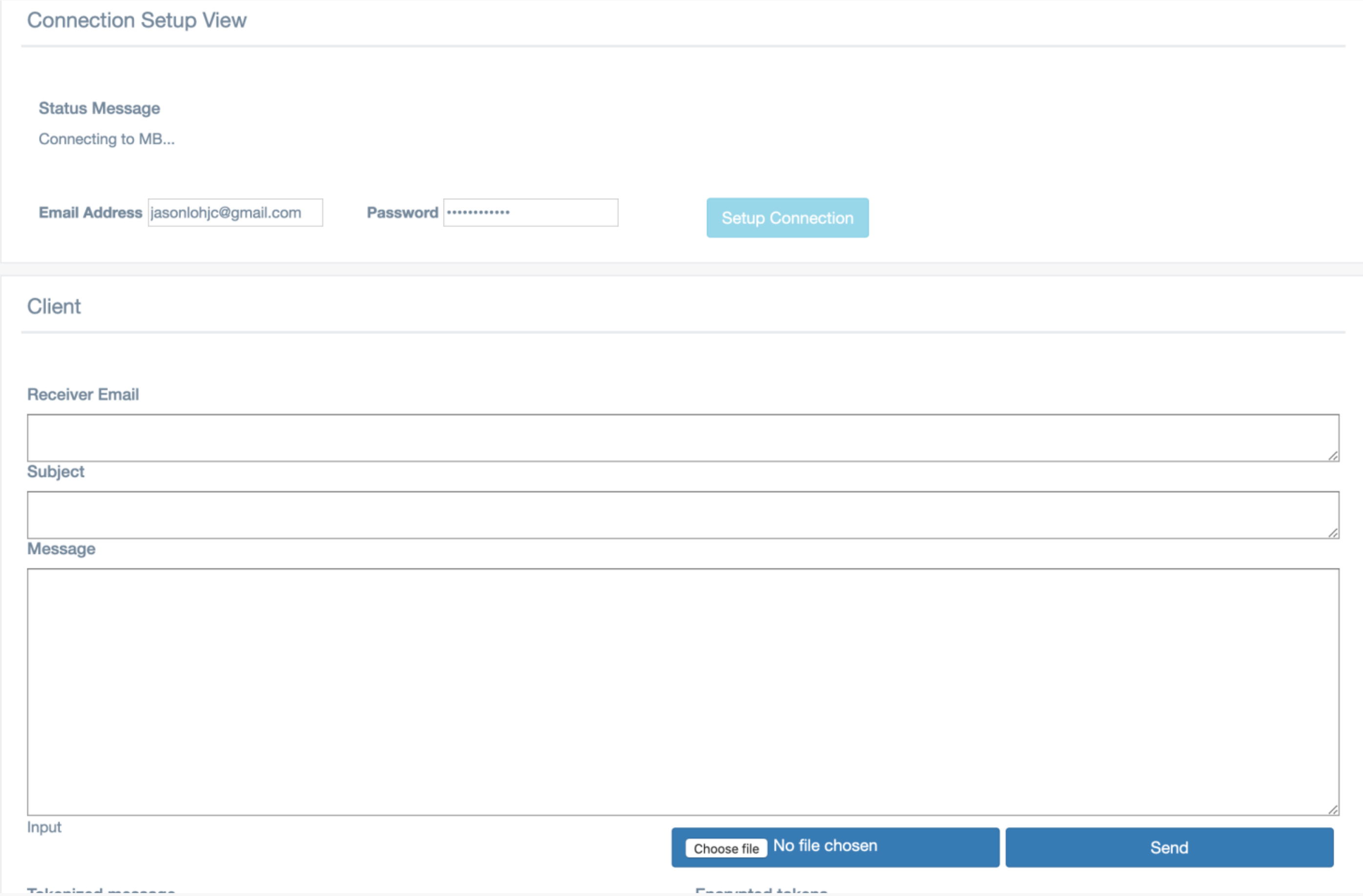
Middlebox – Rules generations
(Done by RuleGenerator)



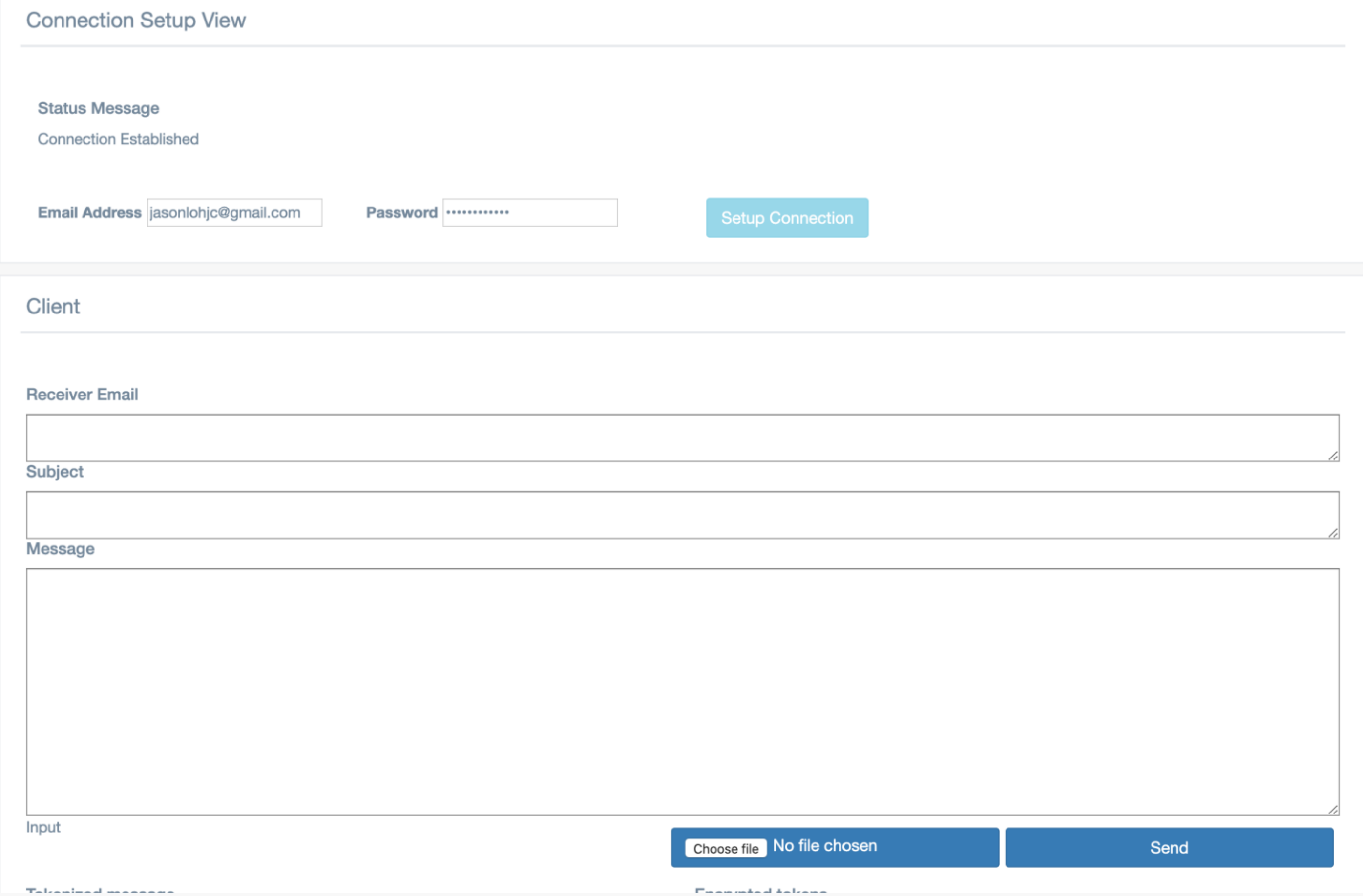
Middlebox – Connection is up



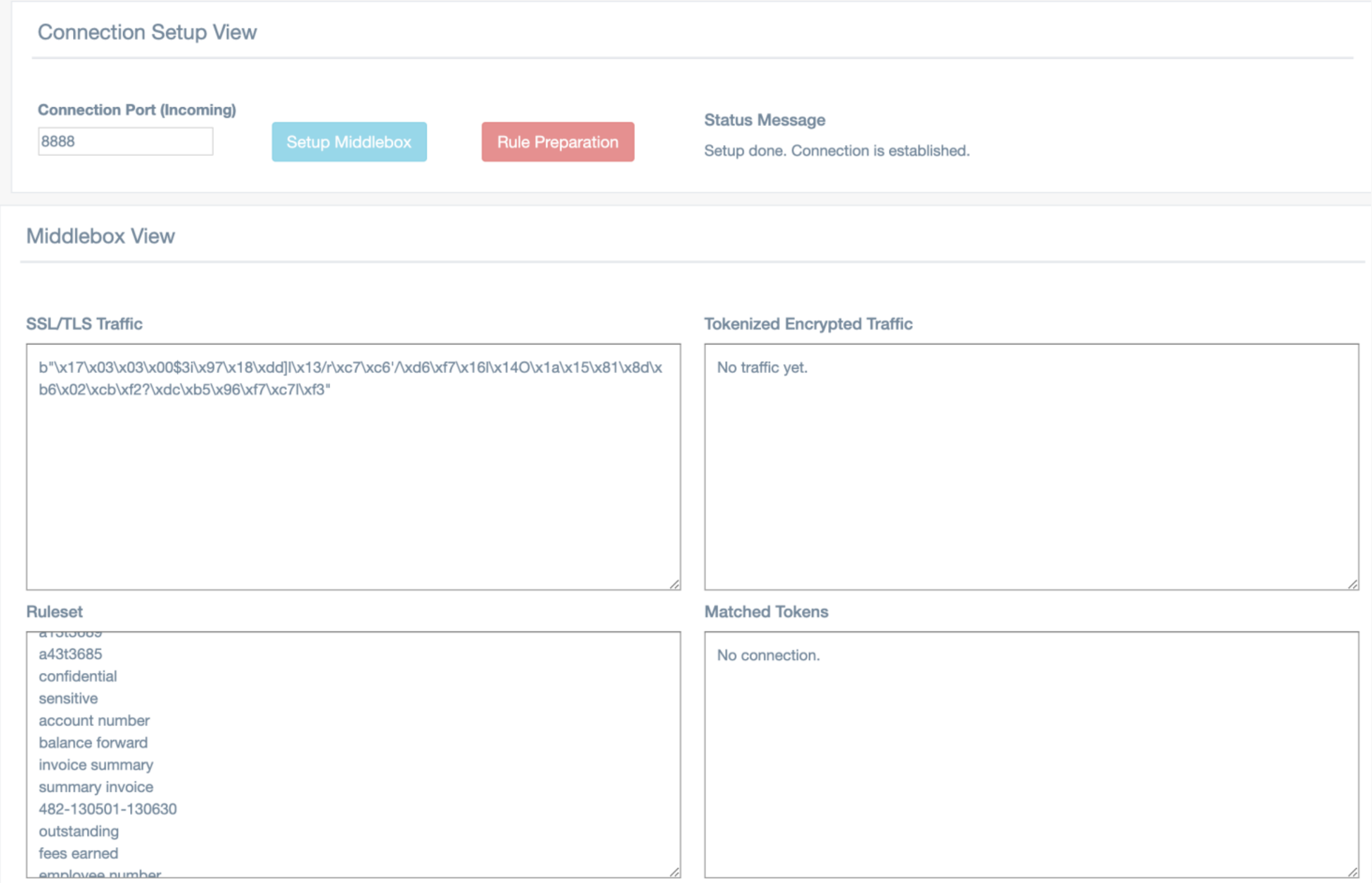
Client – Logging to email account



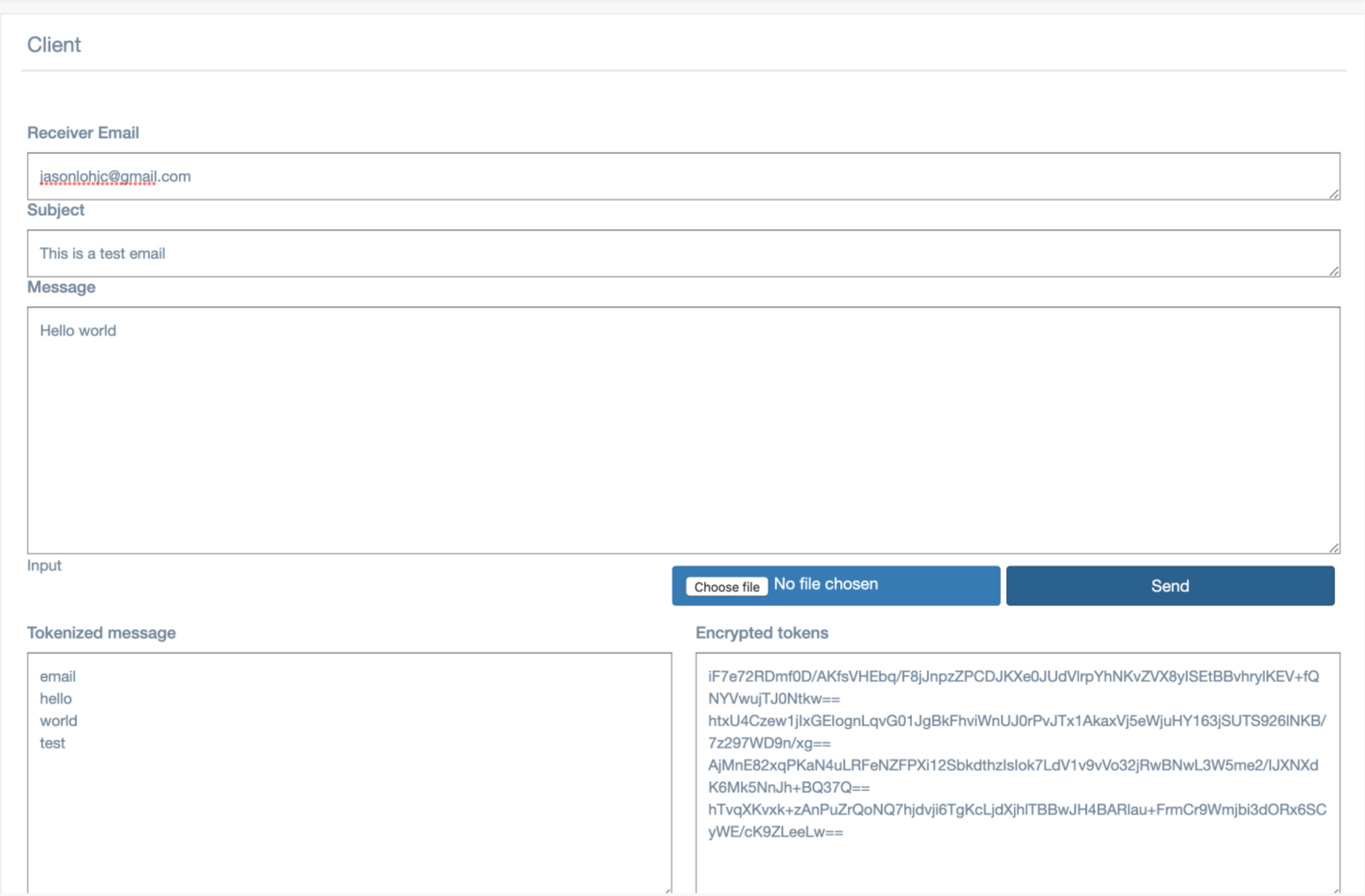
Client – Connection is established with Middlebox to Google



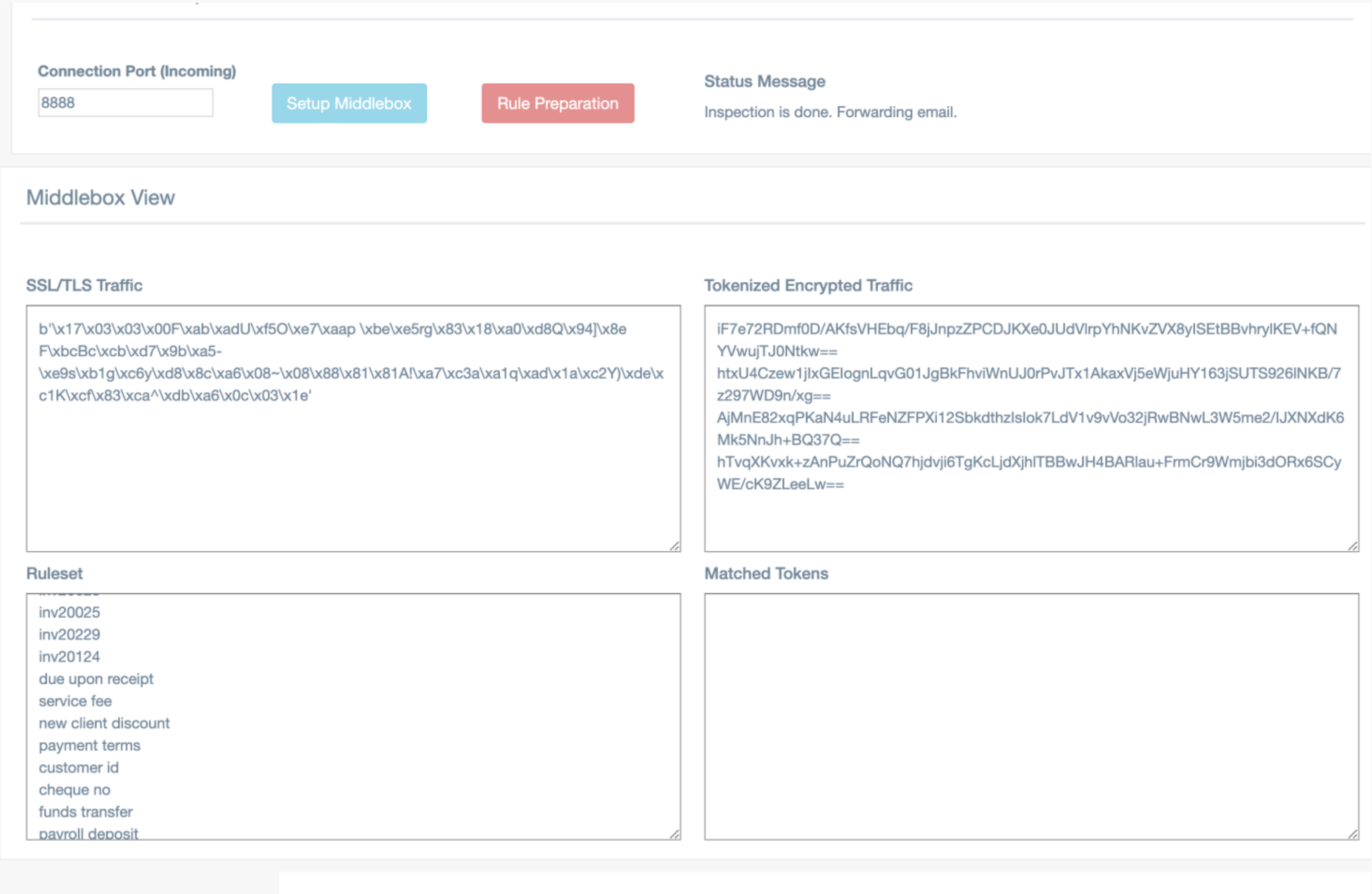
Middlebox – Connection is establish with Client and Google. Note that TLS traffics is forwarded.



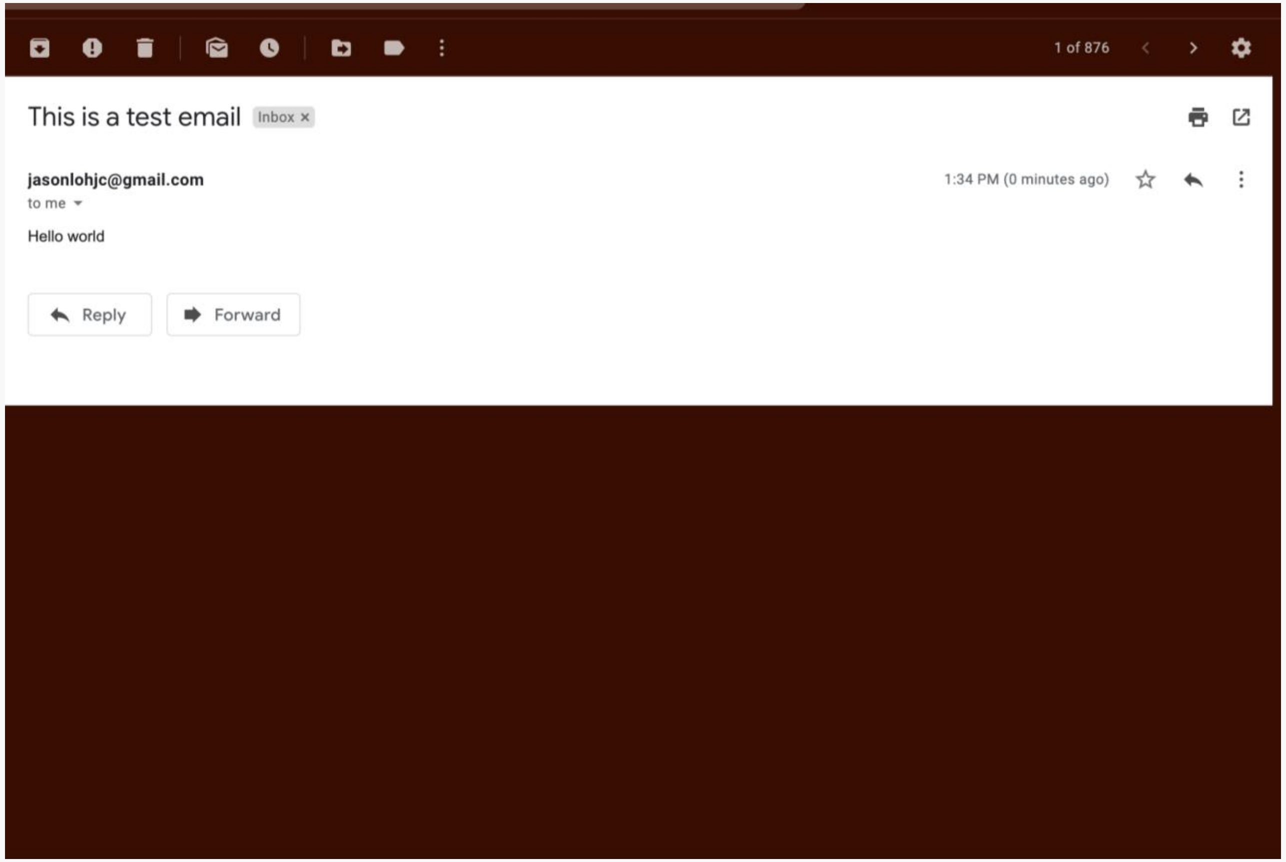
Client – Sends a clean email



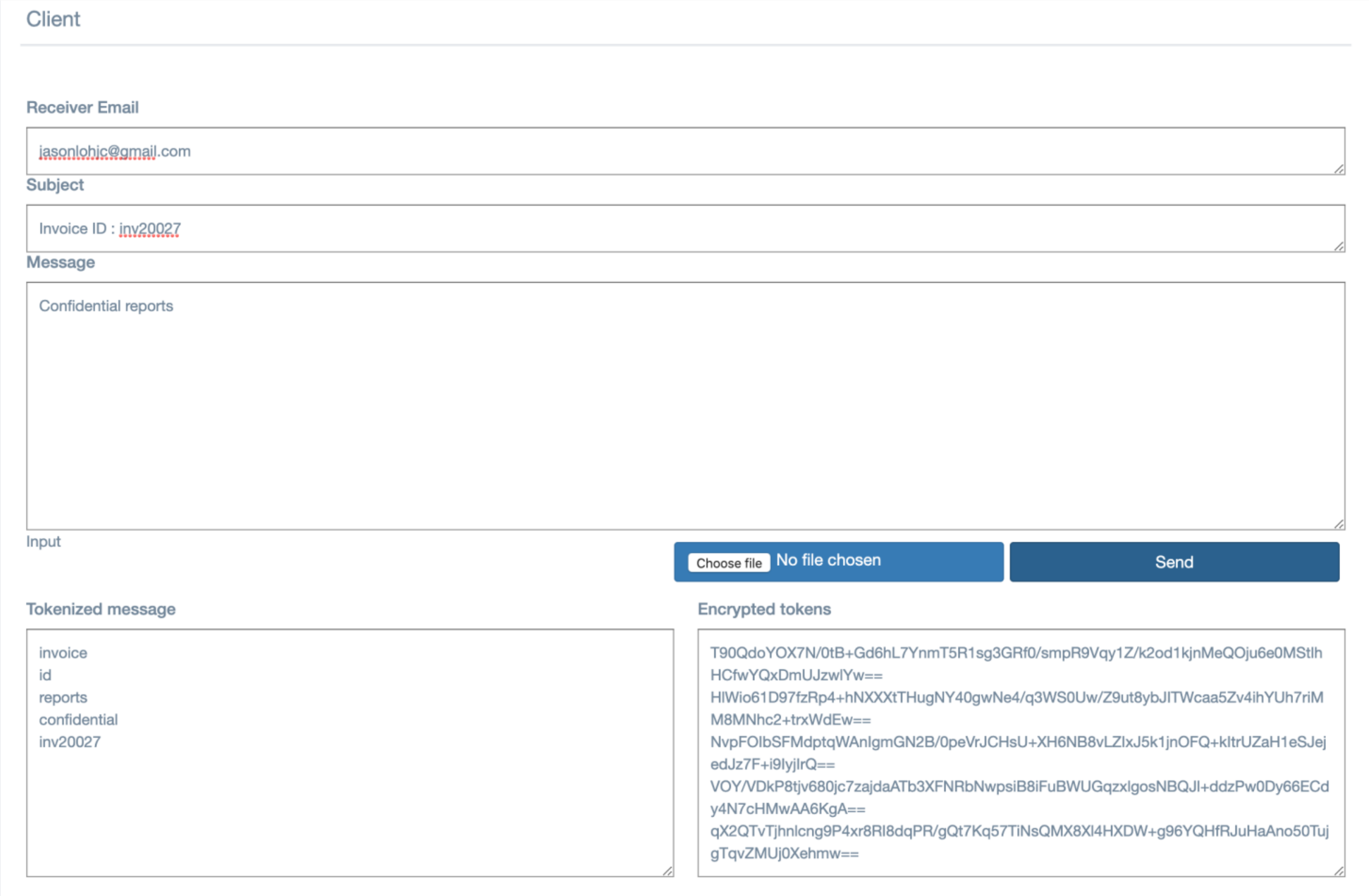
Middlebox – Email is clean. Forward the Google



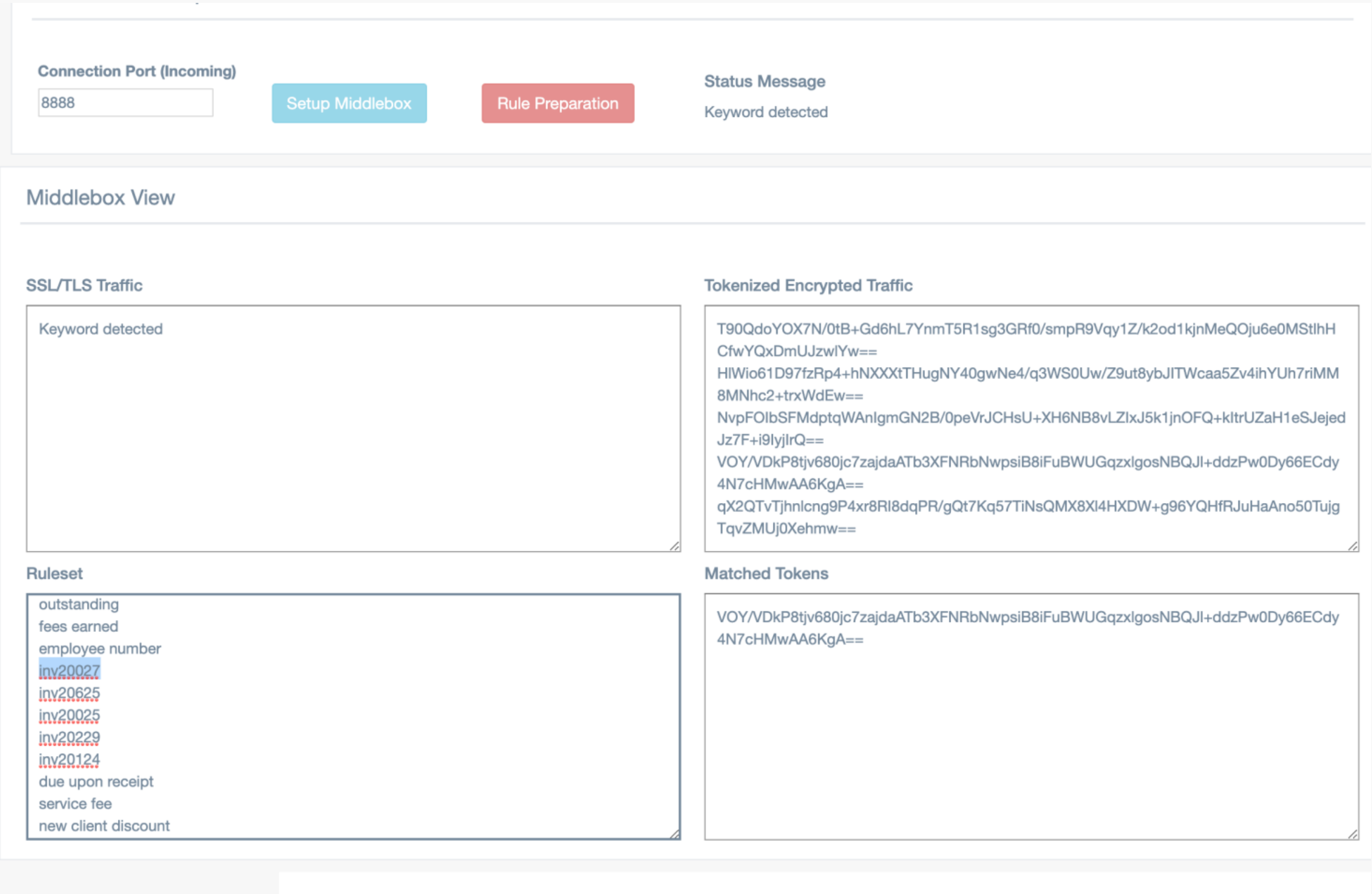
Google - Receive the email



Client – Sends a sensitive email



Middlebox – Detected! Blocked!



Conclusion

Conclusion

- We proposed a new technique called obfuscated rule encryption to minimize the computation and communication overhead during the setup phase while preserving the same properties and privacy requirements as in BlindBox.

- We introduced the idea of session reusable that allows MB to reuse the encrypted rules.

- However, token encryption is roughly 6x slower than BlindBox, so we introduced reusable token encryption to achieve only 3.5x slower in the ideal case.

Q & A

Concrete Scheme

PrivDPI: Setup

MB

RG

$$pk_{rg} = g^{sk_{rg}},$$

$$A = g^{\alpha},$$

$$R_i = g^{r_i \alpha + s_i}$$

$$\sigma_{r_i} = Sign(sk_{rg}, R_i) \qquad i \in \{1, \dots, N\}$$

$$\xleftarrow{\;s_i, (R_i, \sigma_{r_i})\;}$$

1. Rule generator (RG) generates obfuscated rules R_i by blinding all the rules r_i where α and s_i are randomly chosen.

2. RG signs R_i with sk_{rg}, so one can verify under pk_{rg} later.

PrivDPI: Setup (v1)

Client

Primary TLS

MB

Server

$\boldsymbol{sk_c = hash(k_{TLS}) \in Z_r}$

1

$pk_c = g^{sk_c},$
$salt_c$

Check:
$pk_c = ?\, pk_s$
$salt_c = ?\, salt_s$

$pk_s = g^{sk_s},$
$salt_s$

2

$(R_i, \sigma_{r_i})_{i \in \{1,\ldots,N\}}$

Verify$(pk_{rg}, R_i, \sigma_{r_i})$

For every R_i,

$$K_i = (R_i \cdot pk_c)^{sk_c}$$
$$= g^{sk_c r_i \alpha + sk_c s_i + sk_c^2}$$

3

K_i

Check:
$e(K_i, g) = ?\, e(R_i \cdot pk_c, pk_s)$

By using bilinear pairings:

- The involvement of S is highly reduced.
- However, it slows down the setup time.

PrivDPI: Setup (v2)

Client

Primary TLS

MB

Server

Check:

$K_i =? K_i'$

$\boldsymbol{sk_c = hash(k_{TLS}) \in Z_r}$

1

$pk_c = g^{sk_c}$,
$salt_c$ →

Check:
$pk_c =? pk_s$
$salt_c =? salt_s$

← $pk_s = g^{sk_s}$,
$salt_s$

2

← $\left(R_i, \sigma_{r_i}\right)_{i \in \{1,\dots,N\}}$

$\left(R_i, \sigma_{r_i}\right)_{i \in \{1,\dots,N\}}$ →

Verify$(pk_{rg}, R_i, \sigma_{r_i})$
For every R_i,

$$K_i = (R_i \cdot pk_c)^{sk_c}$$
$$= g^{sk_c r_i \alpha + sk_c s_i + sk_c^2}$$

Verify$(pk_{rg}, R_i, \sigma_{r_i})$
For every R_i,

$$K_i' = (R_i \cdot pk_c)^{sk_s}$$
$$= g^{sk_s r_i \alpha + sk_s s_i + sk_s^2}$$

3

K_i →

Check:
$K_i =? K_i'$

← K_i'

PrivDPI: Setup (cont.)

Client

Primary TLS

MB

Server

4

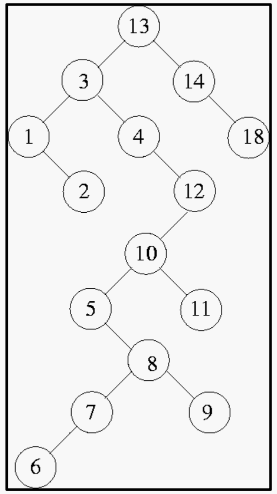
Compute encrypted rule:

$$I_i = \frac{K_i}{(pk_c)^{s_i}} = g^{sk_s r_i \alpha + sk_s^2}$$

For every I_i,

$$C_i = AES_{I_i}(salt_c + ct_{I_i})$$

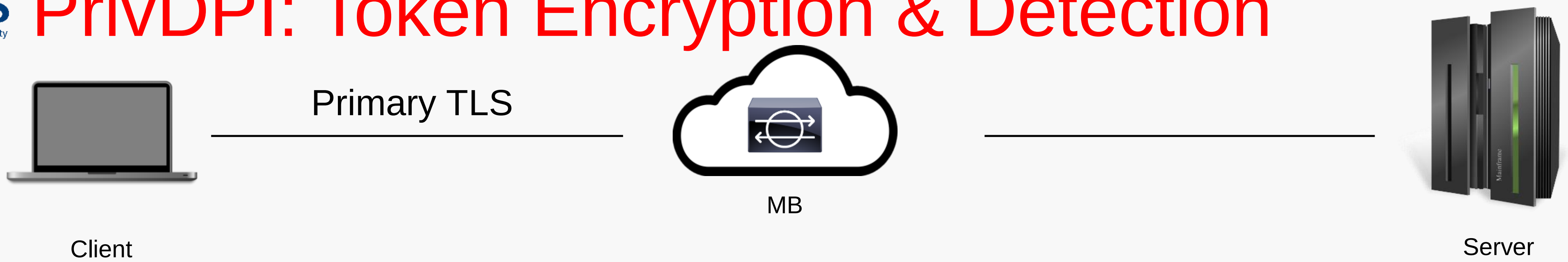
Insert C_i into SearchTree:



A counter table is generated for detection later:

I_i	ct_{I_i}
I_1	0
$\vdots$	0
I_N	0

PrivDPI: Token Encryption & Detection



For any message m,
C tokenizes $m \rightarrow \{t_0, \cdots, t_i\}$

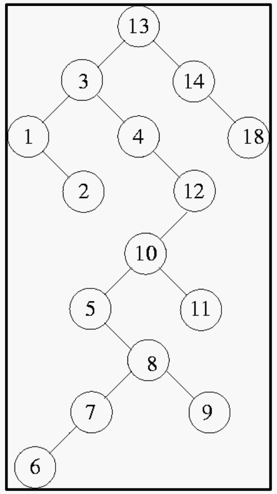
$$P_i = A^{sk_c t_i} \cdot g^{sk_c^2}$$

$$T_i = AES_{P_i}(salt_c + ct_{T_i})$$

Note that for every repeated token, $ct_{T_i} + 1$

1 $\xrightarrow{T_i}$

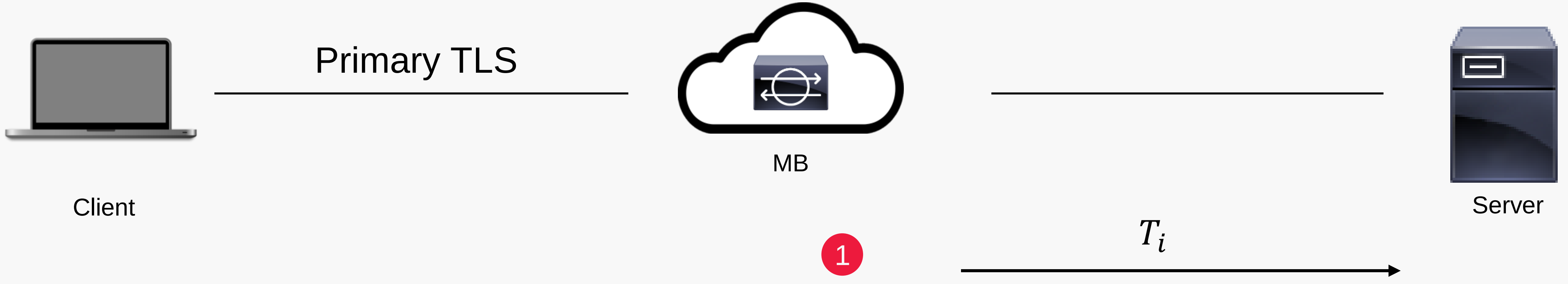
$T_i \quad \lll=\ggg$



For every T_i matches in SearchTree,
Recompute: $C_i = AES_{I_i}(salt_c + ct_{I_i} + 1)$
Update the table and SearchTree

I_i	ct_{I_i}
I_1	0
$\vdots$	$\cancel{0}$ 1
I_N	0

PrivDPI: Token Validation



Receive message m' from TLS,
S tokenizes $m' \rightarrow \{t'_0, \cdots, t'_i\}$

$$P'_i = A^{sk_s t'_i} \cdot g^{sk_s^2}$$

$$T'_i = AES_{P'_i}(salt_c + ct_{T'_i})$$

Note that for every repeated token, $ct_{T'_i} + 1$

$$T_i =? \; T'_i$$

PrivDPI: Setup (Session Reusable)

Client

Primary TLS

MB

Server

sk_c (computed in first session)
$sk_{c2} = hash(k_{TLS}) \in Z_r$ (new session)

1

$pk_{c2} = g^{sk_{c2}}$,
$salt_{c2}$

Check:
$pk_{c2} =? \, pk_{s2}$
$salt_{c2} =? \, salt_{s2}$

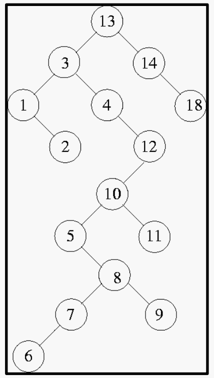
$pk_{s2} = g^{sk_{s2}}$,
$salt_{s2}$

2

MB re-computes every I_i:

$$I_i' = I_i \cdot pk_{c2} = g^{sk_c r_i \alpha + sk_c^2 + sk_{c2}}$$

$$C_i = AES_{I_i'}(salt_{c2} + ct_{I_i'})$$



PrivDPI: Token Encryption & Detection (Session Reusable)

Client

Primary TLS

MB

Server

For any message m,

C tokenizes $m \rightarrow \{t_0, \cdots, t_i\}$

$P_i = A^{sk_c t_i} \cdot g^{sk_c^2} \cdot g^{sk_{c2}}$

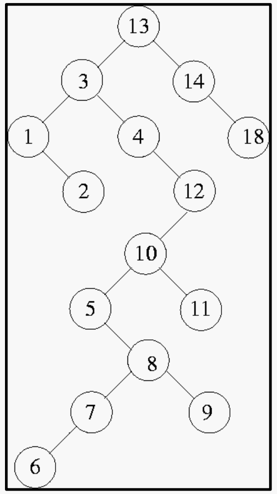
$T_i = AES_{P_i}(salt_c + ct_{T_i})$

Note that for every repeated token, $ct_{T_i} + 1$

1

T_i

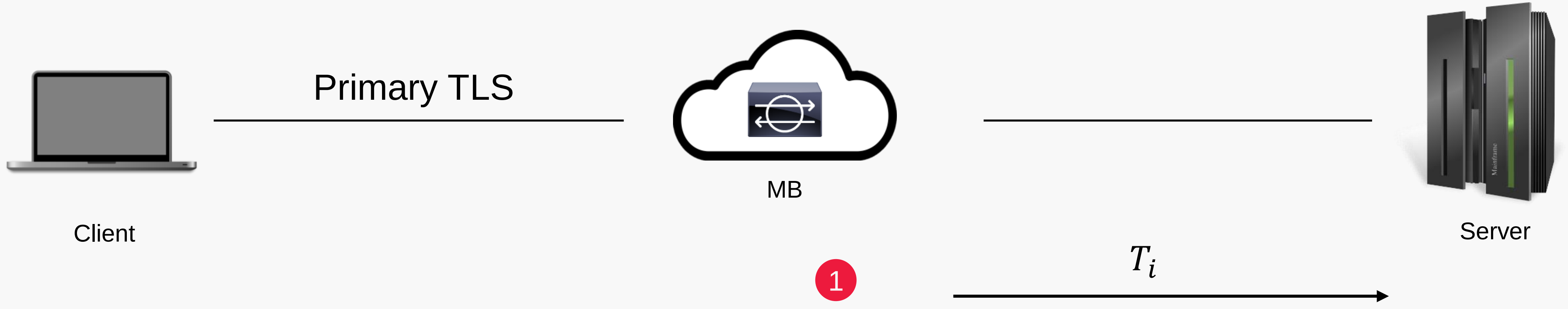
$T_i \quad \lll\!=\!\ggg$



For every T_i matches in SearchTree,
Recompute: $C_i = AES_{I_i}(salt_c + ct_{I_i} + 1)$
Update the table and SearchTree

I_i	ct_{I_i}
I_1	0
$\vdots$	$\cancel{0}$ 1
I_N	0

PrivDPI: Token Validation (Session Reusable)



Receive message m' from TLS,
S tokenizes $m' \rightarrow \{t'_0, \cdots, t'_i\}$

$$P'_i = A^{sk_s t'_i} \cdot g^{sk_s^2} \cdot g^{sk_{s2}}$$

$$T'_i = AES_{P'_i}(salt_c + ct_{T'_i})$$

Note that for every repeated token, $ct_{T'_i} + 1$

$$T_i \; =? \; T'_i$$

Reusable Token Encryption (TE)

The encryption in PrivDPI is more expensive than BlindBox, such that:

BlindBox		PrivDPI	
2 x AES	$AES_k(t) \rightarrow \alpha$ $AES_\alpha(salt + ct)$	1 x Exponentiation 1 x Multiplication 1 x AES	$(A)^t \rightarrow \alpha$ $\alpha \cdot B \rightarrow \beta$ $AES_\beta(salt + ct)$

Pre-compute:

$$A = g^{\alpha \cdot sk_s}$$

$$B = g^{sk_s^2}$$

To improve the performance for C and S, we propose token reuse technique.

Specially for S, those frequently tokens can be reused, e.g. webpage that is surfed frequently.

Reusable TE (Cont.)

For the first connection (1st session), C or S computes each token into such table:

Token	Element
t_A	TE= $g^{\alpha \cdot sk_s \cdot t_A + sk_s^2}$
	Seed = $g^{\alpha \cdot sk_s \cdot t_A + sk_s^2}$
	Counter = 1
	Session = 1
t_B	TE= $g^{\alpha \cdot sk_s \cdot t_B + sk_s^2}$
	Seed = $g^{\alpha \cdot sk_s \cdot t_B + sk_s^2}$
	Counter = 1
	Session = 1

$$AES_{TE}(salt + Counter)$$

Suppose that C or S wants to encrypt t_A again.

As t_A was computed, C or S can retrieve TE=$g^{\alpha \cdot sk_s \cdot t_A + sk_s^2}$ and perform 1 x AES only.

Then C or S updates the table, such that Counter increases by 1.

Token	Element
t_A	TE= $g^{\alpha \cdot sk_s \cdot t_A + sk_s^2}$
	Seed = $g^{\alpha \cdot sk_s \cdot t_A + sk_s^2}$
	Counter = **2**
	Session = 1