Proxy Re-Encryption

Redirect end-to-end encrypted traffic

1. What is the problem with end-to-end encrypted communication?
2. What is Proxy Re-Encryption and how can it solve the problem?
3. What flavours does it come in?
4. How does it work?
5. What are the drawbacks of Proxy Re-Encryption?

Who am I?

Freie Universität Berlin

SSE

- Bachelor in maths **@ Freie Universität Berlin**
- Master in computer science **@ Freie Universität Berlin**
- Master thesis on Proxy Re-Encryption and its uses for electronic mail boxes
- Security Expert **@ SSE**

Teetje Stark

- ✓ Doppelkopf enthusiast
- ✓ Into crypto, sports and games

Who am I?

Agenda

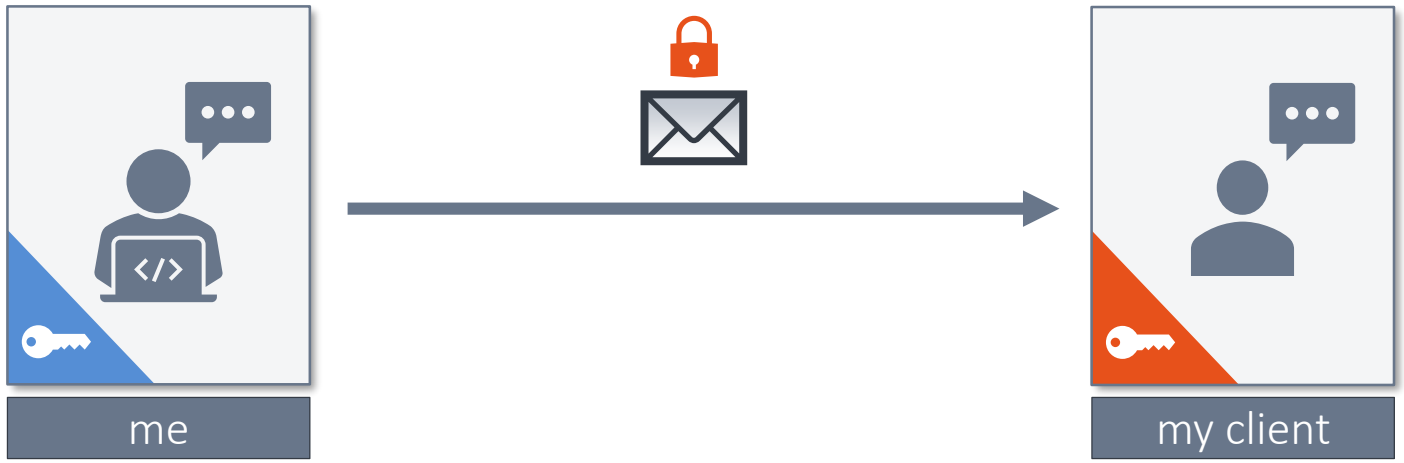
1. What is the problem with end-to-end encrypted communication?
2. What is Proxy Re-Encryption and how can it solve the problem?
3. What flavours does it come in?
4. How does it work?
5. What are the drawbacks of Proxy Re-Encryption?

Reasons to Use End-To-End-Encryption (E2EE)

- Confidentiality
- Integrity
- Authenticity
- Non-repudiation
- Annoy the NSA ;-)

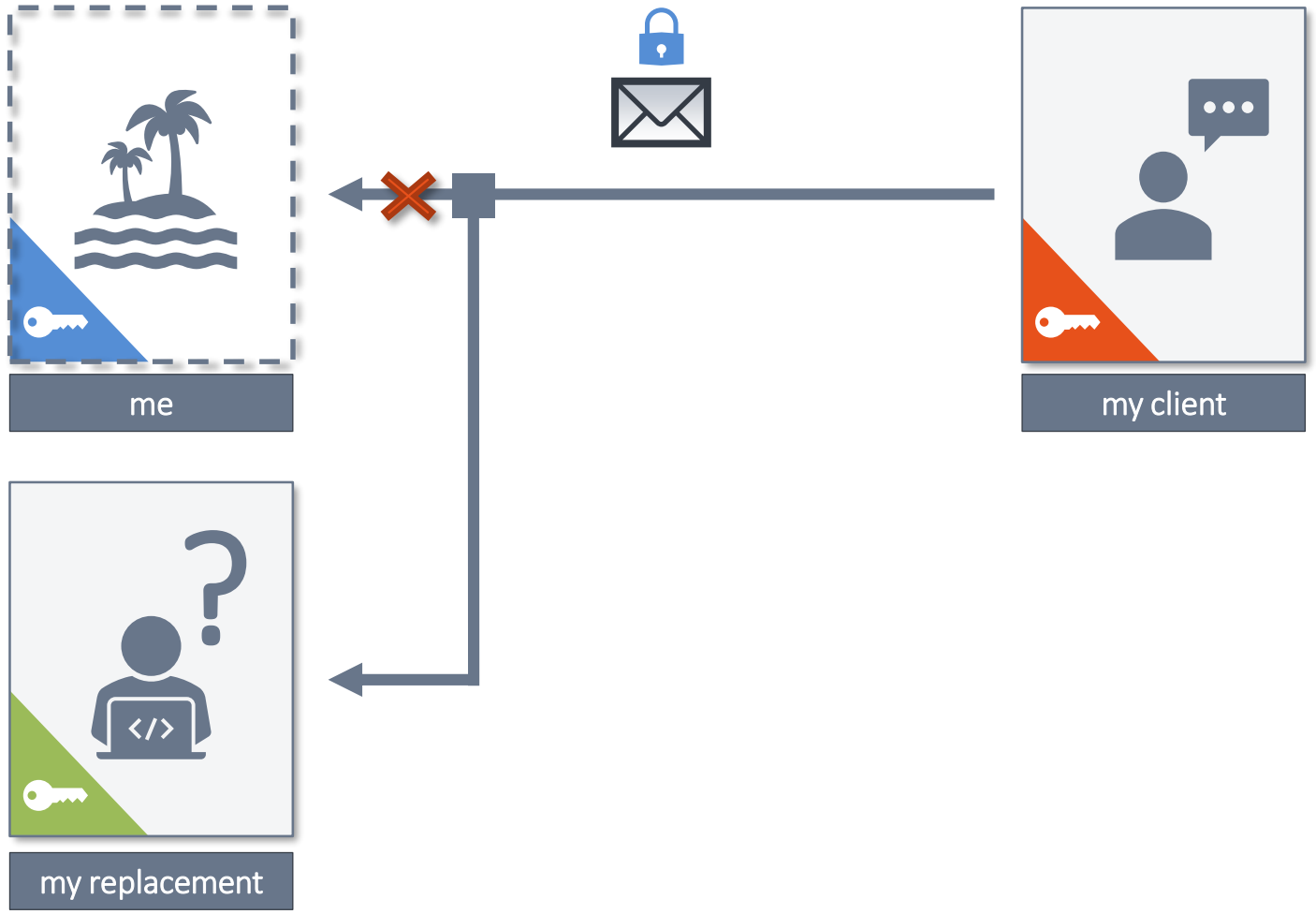
STEP 1: I GO TO WORK

STEP 2: I COMMUNICATE WITH A CLIENT VIA END-TO-END ENCRYPTION

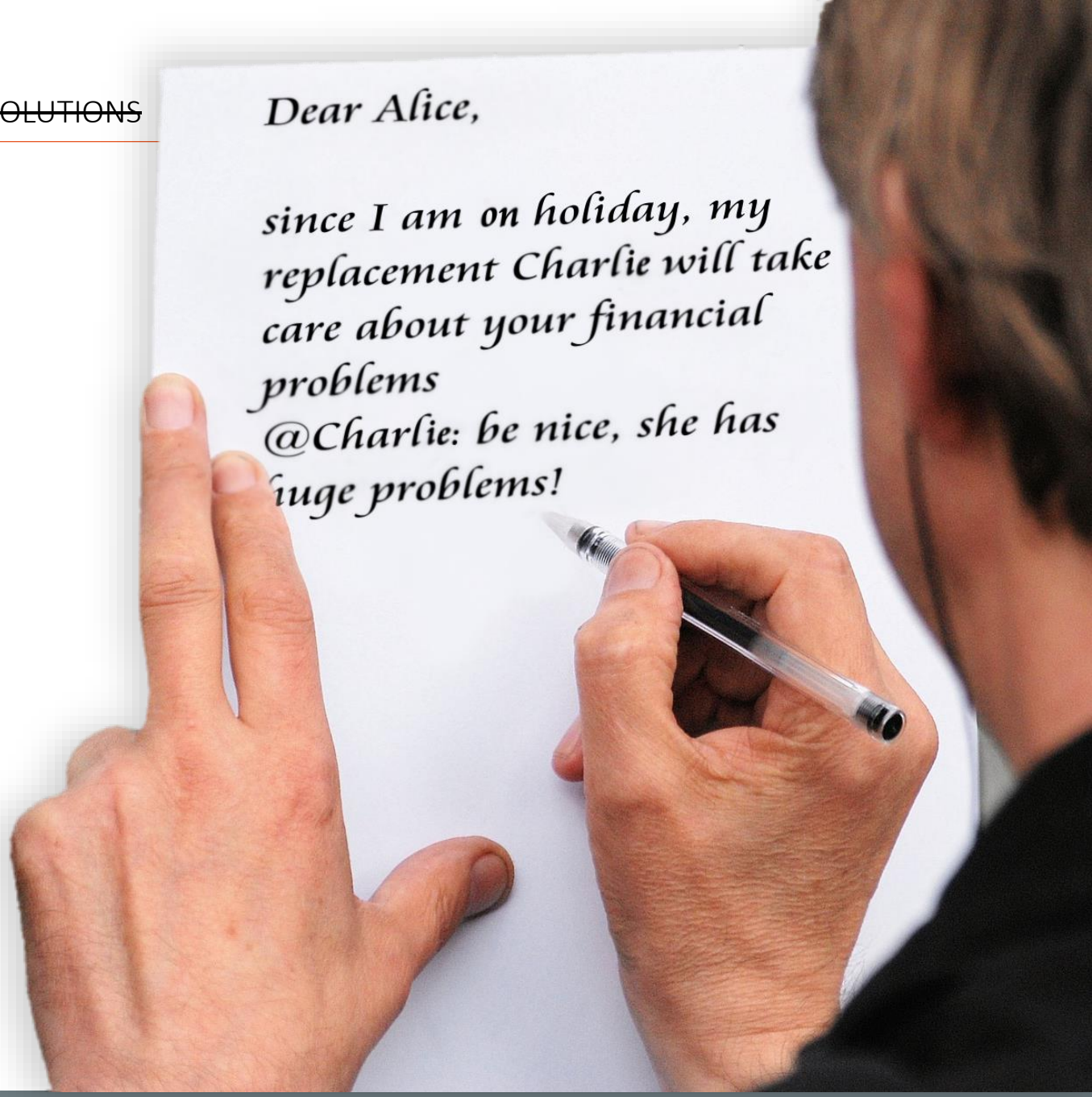


STEP 3: I GO ON VACATION

STEP 4: MY CLIENT WANTS TO TALK, BUT THE REPLACEMENT DOESN'T HAVE THE KEY



JUST DON'T ENCRYPT



SHARE A ~~X~~ THE KEYS

IGNORE UNTIL I'M BACK

Agenda

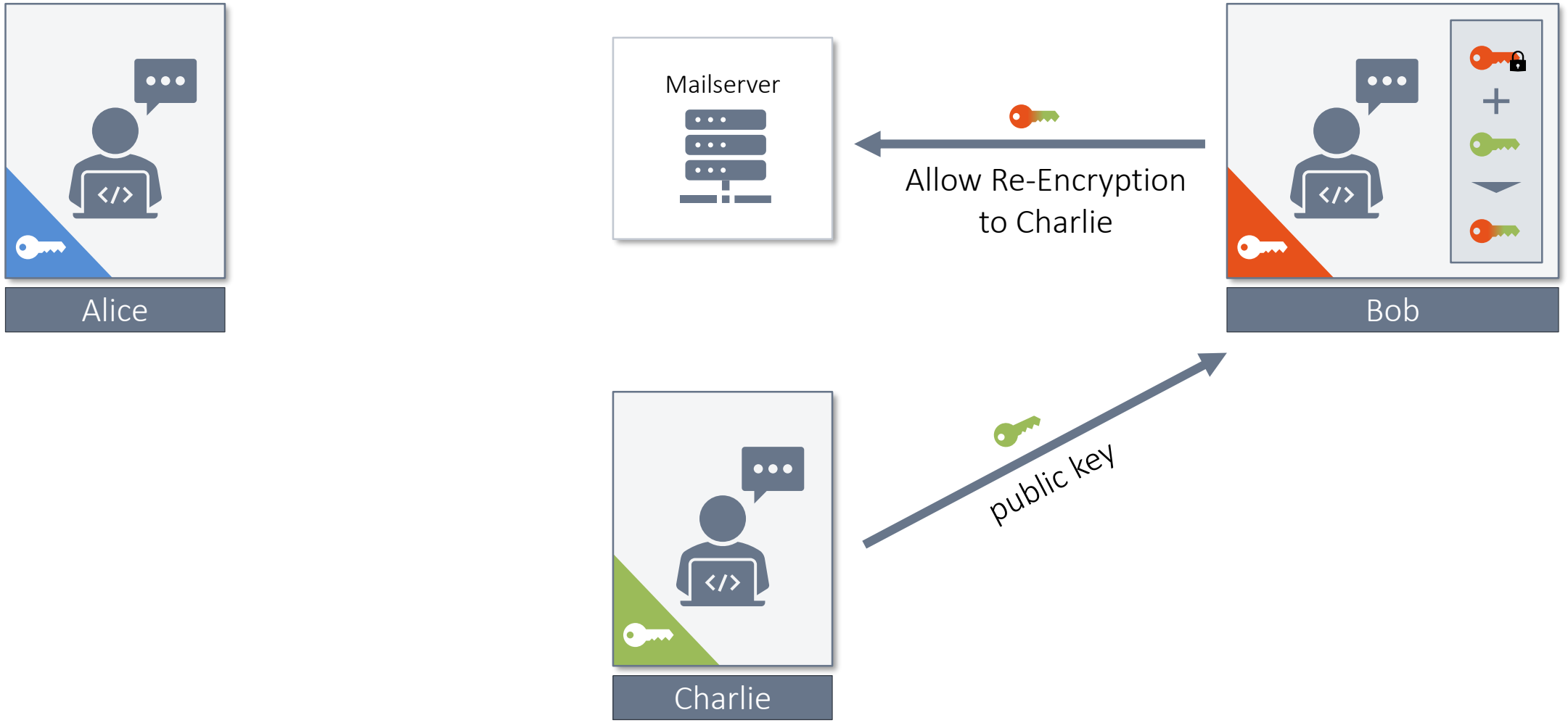
1. What is the problem with end-to-end encrypted communication?
2. **What is Proxy Re-Encryption and how can it solve the problem?**
3. What flavours does it come in?
4. How does it work?
5. What are the drawbacks of Proxy Re-Encryption?

PROXY RE-ENCRYPTION IS A CRYPTOGRAPHIC PRIMITIVE THAT ALLOWS ...

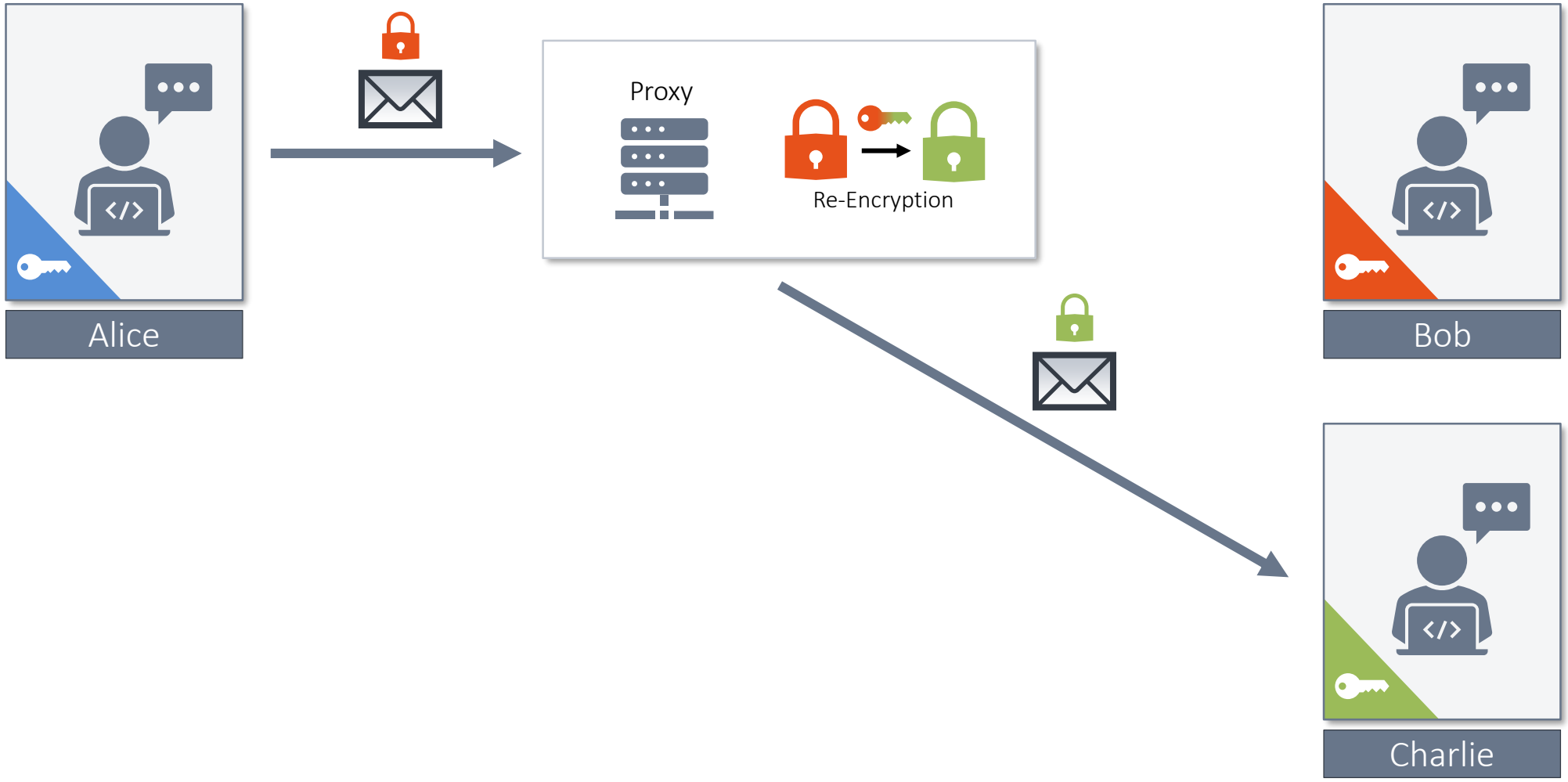
... „FORWARDING“ (RE-ENCRYPTION) OF E2EE CIPHERTEXTS BY A SEMI-TRUSTED PROXY (E.G. MAIL SERVER)

Usual Setup

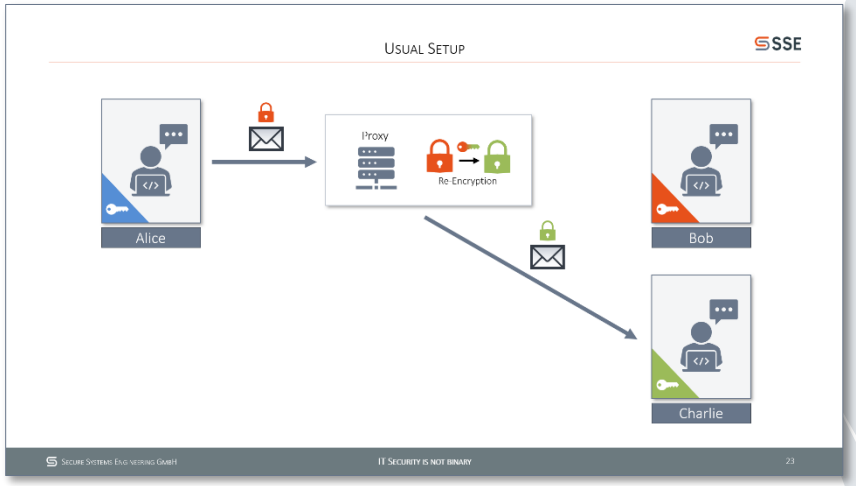
Usual Setup


Alice
Mailserver
Allow Re-Encryption
to Charlie
Bob
Charlie
public key

Usual Setup



Proxy Re-Encryption



- Proxy **never** sees the **plaintext**
- **Only with permission** of Bob can the proxy re-encrypt mails
- Charlie cannot read Bobs mails without the proxies help
- If either Bob or the Proxy are honest, only the messages intended to be forwarded will be forwarded
  - Time-boxed
  - Could be based on additional attributes that are not encrypted (better: attribute-based encryption)

In a company the semi-trusted mail server is a reasonable assumption

Agenda

1. What is the problem with end-to-end encrypted communication?
2. What is Proxy Re-Encryption and how can it solve the problem?
3. **What flavours does it come in?**
4. How does it work?
5. What are the drawbacks of Proxy Re-Encryption?

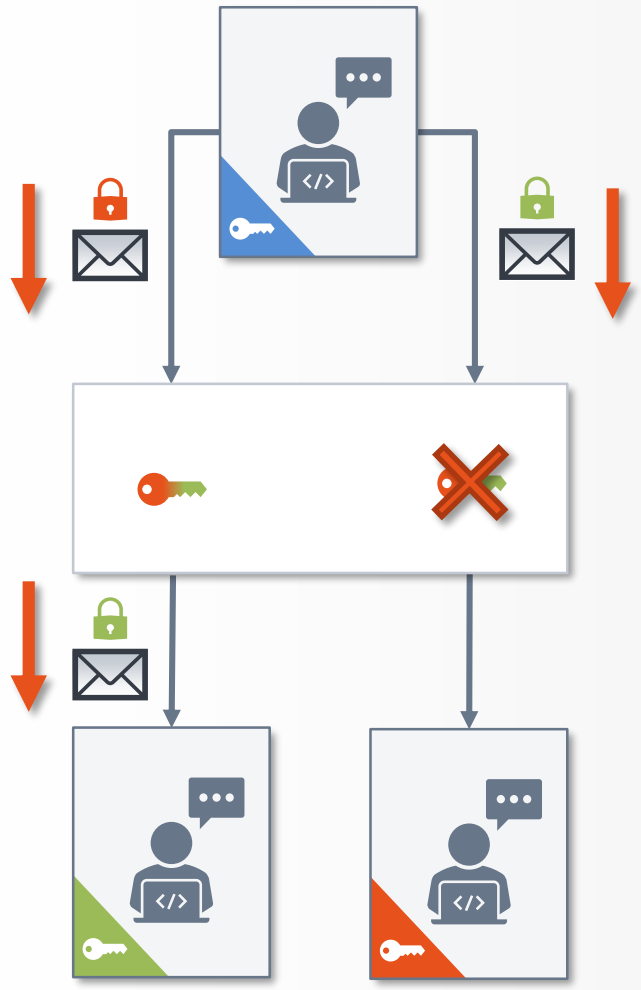
There are multiple types of Proxy Re-Encryption

SSE

	vs.	
UNIDIRECTIONAL		BIDIRECTIONAL
		
SINGLE-USE	vs.	MULTI-USE
		
CPA-SECURE	vs.	CCA-SECURE
		
WITHOUT MASTER SECRET SECURITY	vs.	WITH MASTER SECRET SECURITY
		

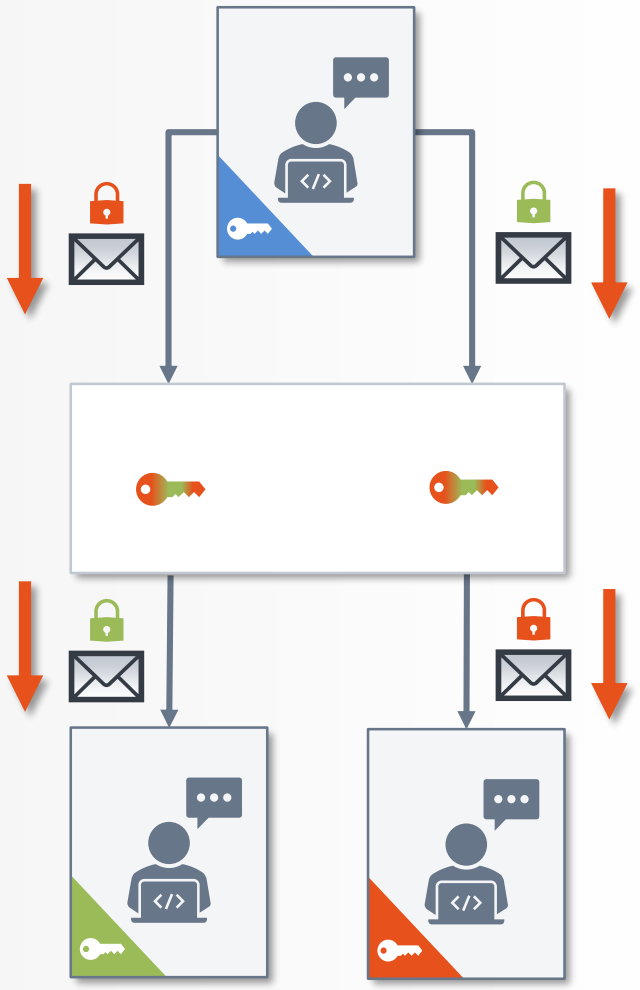
There are multiple types of Proxy Re-Encryption

UNIDIRECTIONAL vs. BIDIRECTIONAL



re-encryption key can only be used in one direction

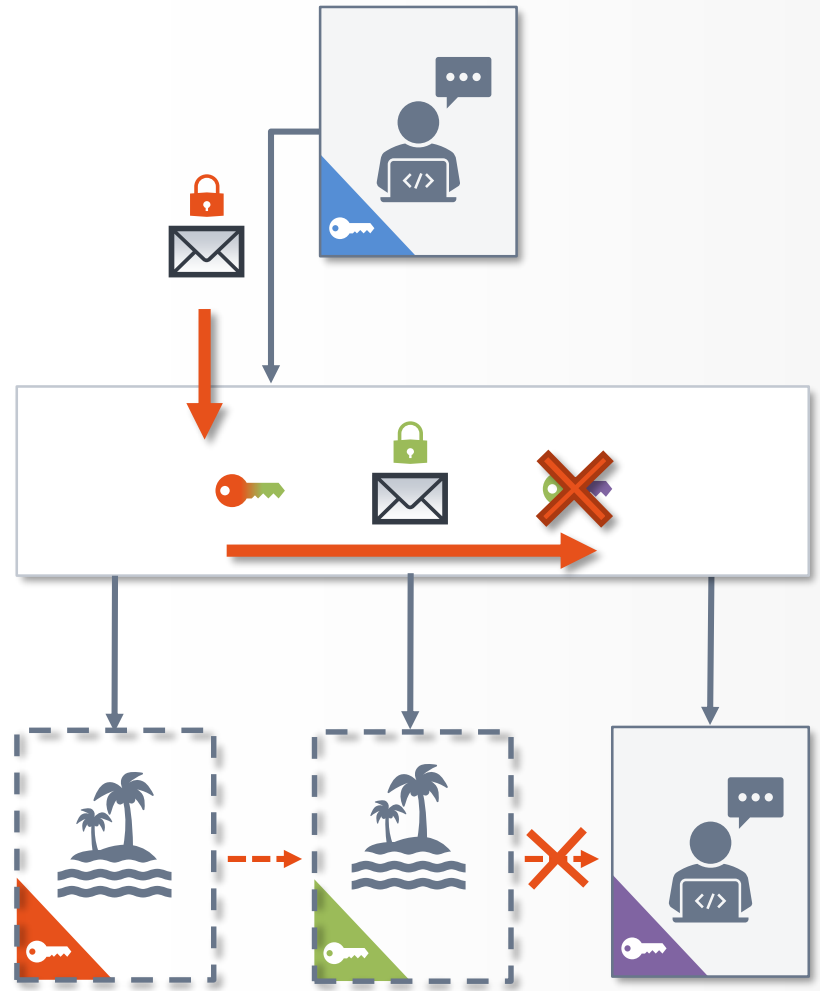
re-encryption key can be used in both directions



There are multiple types of Proxy Re-Encryption

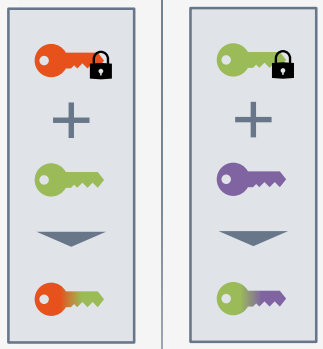
SSE

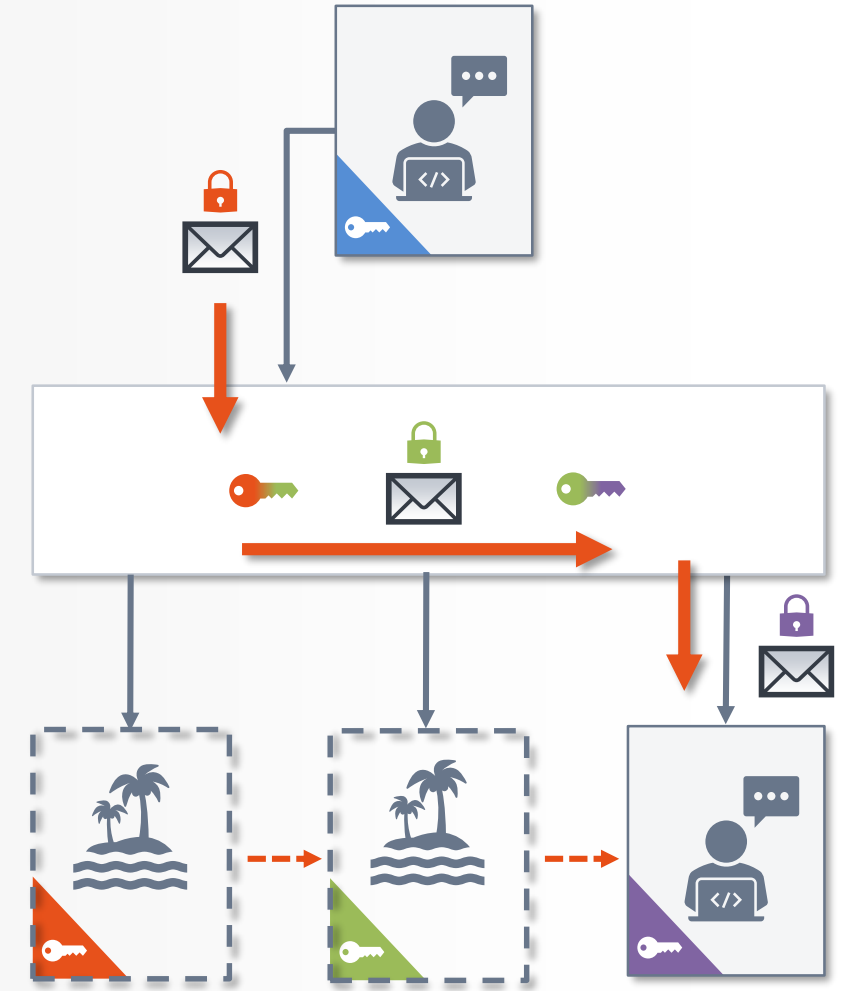
SINGLE-USE vs. MULTI-USE



re-encryption can only be done once

re-encryption can be done multiple times





There are multiple types of Proxy Re-Encryption

UNIDIRECTIONAL vs. BIDIRECTIONAL

SINGLE-USE vs. MULTI-USE

CPA-SECURE vs. CCA-SECURE

WITHOUT MASTER SECRET SECURITY vs. WITH MASTER SECRET SECURITY

Agenda

1. What is the problem with end-to-end encrypted communication?
2. What is Proxy Re-Encryption and how can it solve the problem?
3. What flavours does it come in?
4. **How does it work?**
5. What are the drawbacks of Proxy Re-Encryption?

What is the most useful attribute?

Unidirectionality

How is it (usually) achieved?

Bilinear maps

What are bilinear maps?

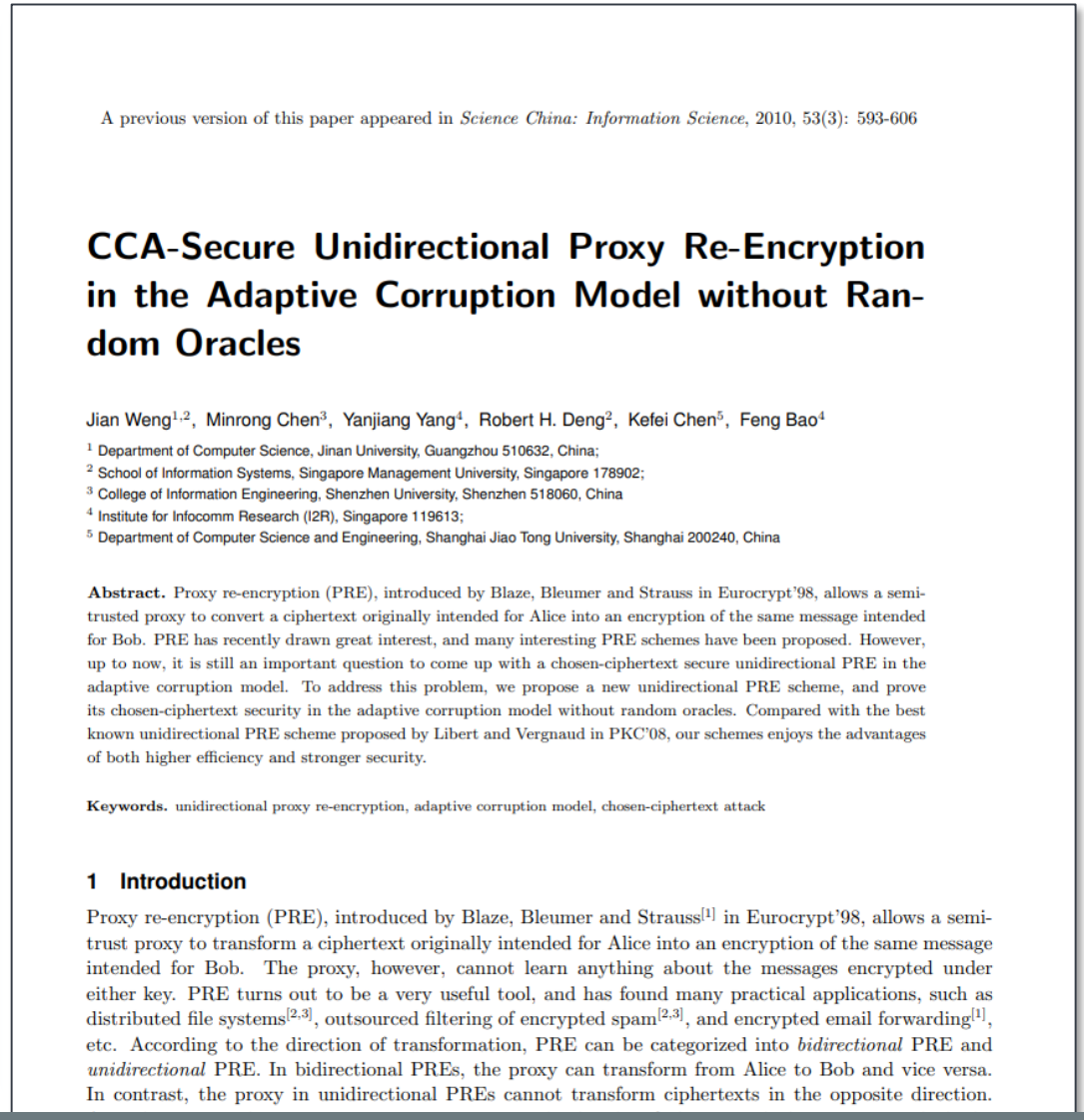
Bilinear maps are maps $\mathbf{e}$: $G_1 \times G_2 \rightarrow G_3$, usually $\mathbf{G} \coloneqq G_1 = G_2$, with G_i cyclic groups of prime order p, which are:

- **Bilinear:** $\forall\, g, h \in G, a, b \in \mathbb{Z}_p$: $e(g^a, h^b) = e(g, h)^{ab}$
- **Non-degenerate:** $\exists g, h \in G$: $e(g, h) \neq \mathbf{1}_{G_3}$

Weng et al.

CCA-Secure Unidirectional Proxy Re-Encryption in the Adaptive Corruption Model without Random Oracles [1]

[1] https://eprint.iacr.org/2010/265.pdf



A previous version of this paper appeared in *Science China: Information Science*, 2010, 53(3): 593-606

CCA-Secure Unidirectional Proxy Re-Encryption in the Adaptive Corruption Model without Random Oracles

Jian Weng[1,2], Minrong Chen[3], Yanjiang Yang[4], Robert H. Deng[2], Kefei Chen[5], Feng Bao[4]

[1] Department of Computer Science, Jinan University, Guangzhou 510632, China;
[2] School of Information Systems, Singapore Management University, Singapore 178902;
[3] College of Information Engineering, Shenzhen University, Shenzhen 518060, China
[4] Institute for Infocomm Research (I2R), Singapore 119613;
[5] Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, China

Abstract. Proxy re-encryption (PRE), introduced by Blaze, Bleumer and Strauss in Eurocrypt'98, allows a semi-trusted proxy to convert a ciphertext originally intended for Alice into an encryption of the same message intended for Bob. PRE has recently drawn great interest, and many interesting PRE schemes have been proposed. However, up to now, it is still an important question to come up with a chosen-ciphertext secure unidirectional PRE in the adaptive corruption model. To address this problem, we propose a new unidirectional PRE scheme, and prove its chosen-ciphertext security in the adaptive corruption model without random oracles. Compared with the best known unidirectional PRE scheme proposed by Libert and Vergnaud in PKC'08, our schemes enjoys the advantages of both higher efficiency and stronger security.

Keywords. unidirectional proxy re-encryption, adaptive corruption model, chosen-ciphertext attack

1 Introduction

Proxy re-encryption (PRE), introduced by Blaze, Bleumer and Strauss[1] in Eurocrypt'98, allows a semi-trust proxy to transform a ciphertext originally intended for Alice into an encryption of the same message intended for Bob. The proxy, however, cannot learn anything about the messages encrypted under either key. PRE turns out to be a very useful tool, and has found many practical applications, such as distributed file systems[2,3], outsourced filtering of encrypted spam[2,3], and encrypted email forwarding[1], etc. According to the direction of transformation, PRE can be categorized into *bidirectional* PRE and *unidirectional* PRE. In bidirectional PREs, the proxy can transform from Alice to Bob and vice versa. In contrast, the proxy in unidirectional PREs cannot transform ciphertexts in the opposite direction.

Proxy Re-Encryption

Second-level encryption
Create a ciphertext that can be forwarded

$$Enc_2(pk_i = g^{x_i}, m):$$
$$r \xleftarrow{\$} \mathbb{Z}_p^*$$
$$C_1 = g_1^r$$
$$C_2 = pk_i^r$$
$$K = Z^r$$
$$C_3 = [F(K, C_1)]_{l_1} \parallel ([F(K, C_1)])^{l_2} \oplus m$$
$$t \xleftarrow{\$} \mathbb{Z}_p^*$$
$$h = H(C_1, C_3)$$
$$C_4 = \left(u^h v^t w\right)^r$$
$$\text{Return } CT_i = (t, C_1, C_2, C_3, C_4)$$

First-level encryption
Create a ciphertext that can <u>not</u> be forwarded

$$Enc_1(pk_i = g^{x_i}, m):$$
$$r \xleftarrow{\$} \mathbb{Z}_p^*$$
$$C_1 = g_1^r$$
$$C_2' = e(pk_i, g)^r$$
$$K = Z^r$$
$$C_3 = [F(K, C_1)]_{l_1} \parallel ([F(K, C_1)])^{l_2} \oplus m$$
$$t \xleftarrow{\$} \mathbb{Z}_p^*$$
$$h = H(C_1, C_3)$$
$$C_4 = \left(u^h v^t w\right)^r$$
$$\text{Return } CT_i' = (t, C_1, C_2', C_3, C_4)$$

Re-Encryption function
Transform second-level ciphertext to first-level ciphertext

$$ReEnc\left(rk_{i\to j} = g^{x_j/x_i}, CT_i\right): *$$
$$(t, C_1, C_2, C_3, C_4) \leftarrow CT_i$$
$$C_2' = e\left(C_2, rk_{i\to j}\right)$$
$$\text{Return } CT_i' = (t, C_1, C_2', C_3, C_4)$$

- Note that $rk_{i\to j} = g^{x_j/x_i}$ is the re-encryption key, which can be constructed from public key $pk_j = g^{x_j}$ of J and private key x_i of I
- It cannot be generated from g^{x_i} and x_j, i.e. $rk_{j\to i} = g^{x_i/x_j} \neq rk_{i\to j} = g^{x_j/x_i}$
  → Thus the scheme is unidirectional

* Leaving out all required validity checks

F is a pseudo-random function, H is a collision-resistant * hash function

1. What is the problem with end-to-end encrypted communication?
2. What is Proxy Re-Encryption and how can it solve the problem?
3. What flavours does it come in?
4. How does it work?
5. **What are the drawbacks of Proxy Re-Encryption?**

Drawbacks of Proxy Re-Encryption

Mainly based on bilinear maps

- Security assumption is often some weird and scheme-specific adaptation of the bilinear version of the decisional Diffie-Hellman problem
- Bilinear maps are slow[1]

[1] https://crypto.stanford.edu/pbc/times.html (C library) has timing on some pairings ranging from roughly the time of one 1024-bit RSA decryption to 60 times as much time. Note that encryption or decryption with Proxy Re-Encryption may need multiple pairings.

Can someone distinguish between:

$$\left(g, g^{1/a}, g^{a}, g^{(a^2)}, g^{b}, e(g,g)^{b/a^2}\right)$$

and:

$$\left(g, g^{1/a}, g^{a}, g^{(a^2)}, g^{b}, e(g,g)^{c}\right)$$

where $a, b, c \xleftarrow{\$} \mathbb{Z}_p^*$ (3-weak decisional bilinear Diffie-Hellman inversion)?

[1] https://crypto.stanford.edu/pbc/times.html (C library) has timing on some pairings ranging from roughly the time of one 1024-bit RSA decryption to 60 times as much time. Note that encryption or decryption with Proxy Re-Encryption may need multiple pairings.

Drawbacks of Proxy Re-Encryption

Mainly based on bilinear maps

- Security assumption is often some weird and scheme-specific adaptation of the bilinear version of the decisional Diffie-Hellman problem
- Bilinear maps are slow[1]

Proxy

Still requires a certain level of trust in the proxy

[1] https://crypto.stanford.edu/pbc/times.html (C library) has timing on some pairings ranging from roughly the time of one 1024-bit RSA decryption to 60 times as much time. Note that encryption or decryption with Proxy Re-Encryption may need multiple pairings.

Thank you!

If you have questions, just ask my holiday replacement ... ;-)