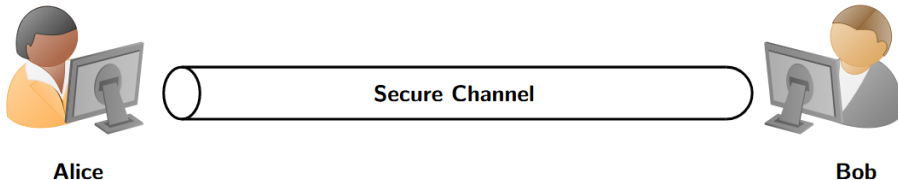




HOW FAST DO YOU HEAL ?

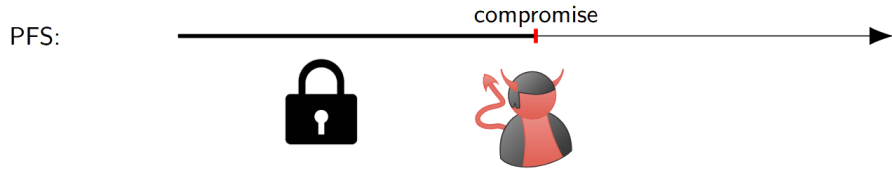
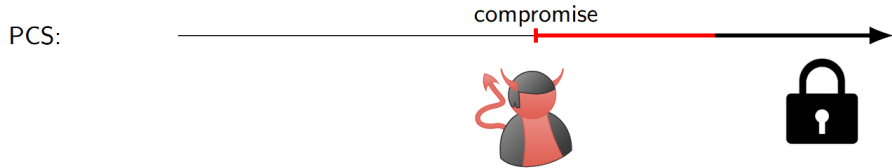
A taxonomy for post-compromise security in secure-channel establishment

ASYNCHRONOUS MESSAGING PROTOCOLS



- Asynchronicity
- Mutual Authentication
- Perfect Forward Secrecy (PFS)
- **Post-Compromise Security (PCS)**
- ...

PFS (EC'89) AND PCS (CSF'16)



STAGE DEFINITION



Chain 1

1. Hello
2. How are you?

⋮

- Stage (1,1): "Hello"
- Stage (2,1): "How are you?"
- Stage (1,2): "Good, and you?"



Chain 2

1. Good, and you?

⋮



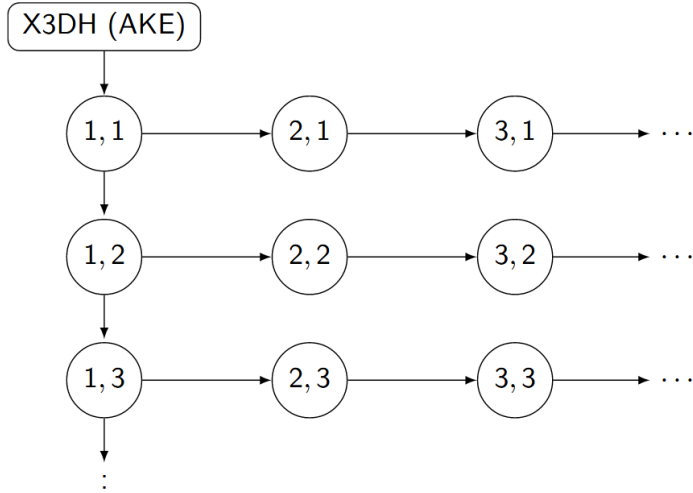
Chain 3

1. Great!

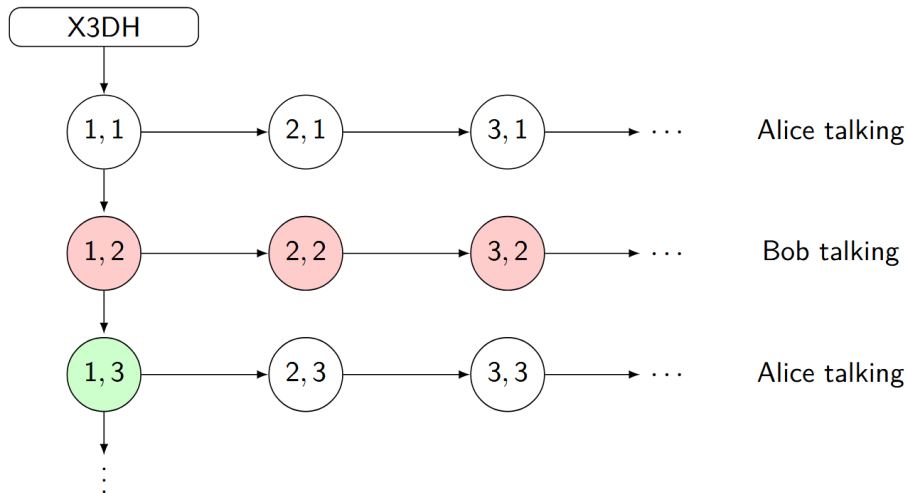
⋮

- Stage (x, y) : x^{th} message of chain y .

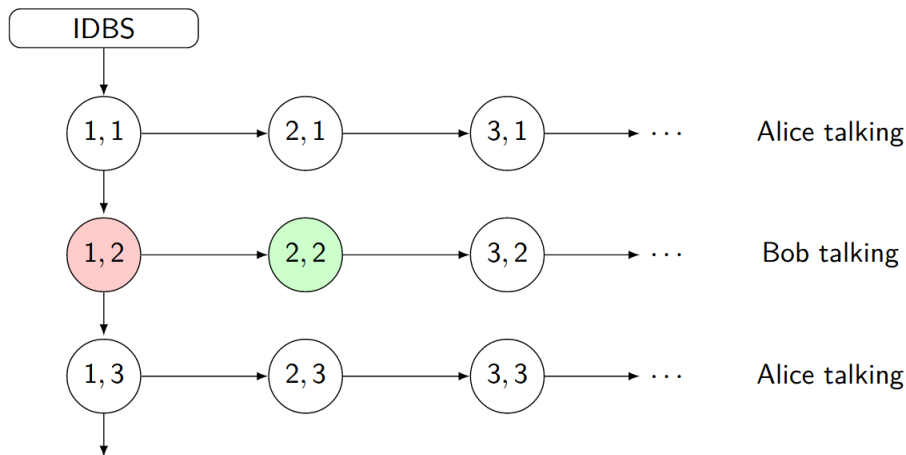
STAGE FLOW OF SIGNAL



IN TERM OF PCS



LOOKING AT SAID (EUROS&P'19)



SAID is $(1,0)$ -PCS

SIGNAL VS SAID?



SAID is better than Signal for PCS?

Ok, we cheated a little

- Key hierarchy
- Identity-based
- Key Distribution Authority vs Database

OUR GOAL IN THIS WORK



- Define the adversary/attacker for a better model
- Quantify healing speed
- Compare different protocols through PCS



We will consider the number of stages *lost* after a compromise

OUTLINE



- Description of Signal
- Example of attacks
- Marshal and Samurai protocols
- Our model to quantify healing
- Applying PCS for 5G?



DESCRIPTION OF SIGNAL

×

def: something (such as a sound, gesture, or object) that conveys notice or warning



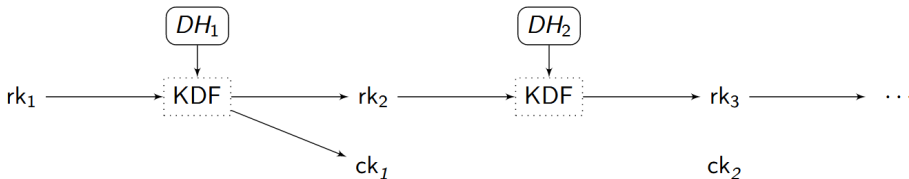
RATCHET ALGORITHMS



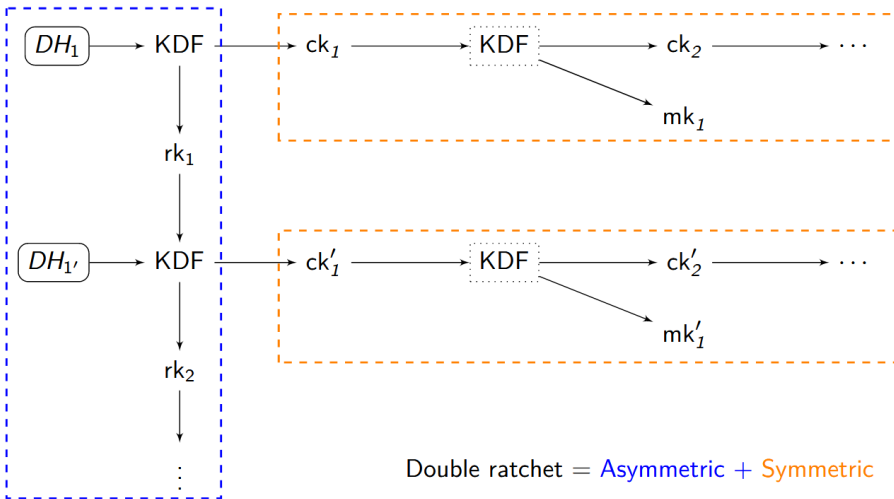
Symmetric Ratchet (PFS)



Asymmetric Ratchet (PCS)

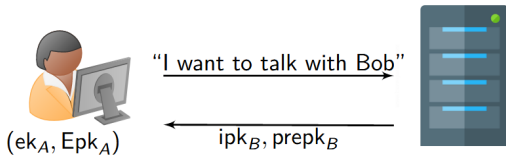


DOUBLE RATCHET ALGORITHM



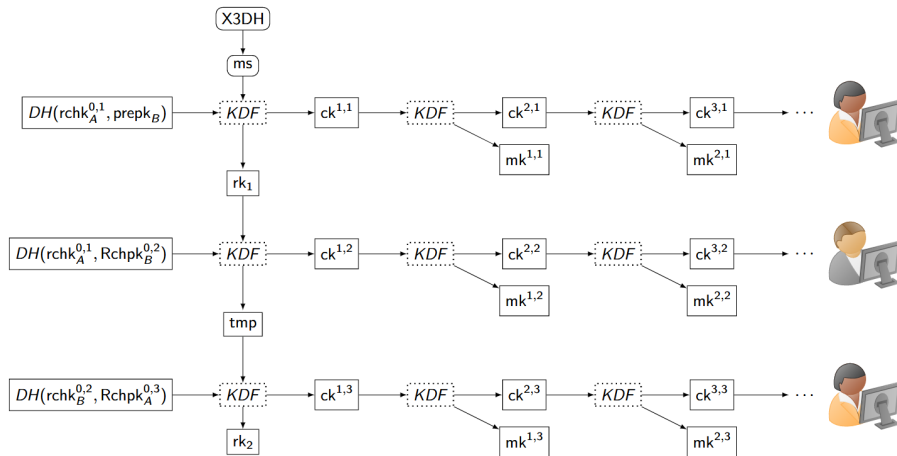
Double ratchet = Asymmetric + Symmetric

REGISTRATION - SETUP



$$\text{X3DH: } ms := (prepk_B)^{ik_A} \parallel (ipk_B)^{ek_A} \parallel (prepk_B)^{ek_A}$$

KEY SCHEDULE OF SIGNAL





EXAMPLE OF ATTACKS

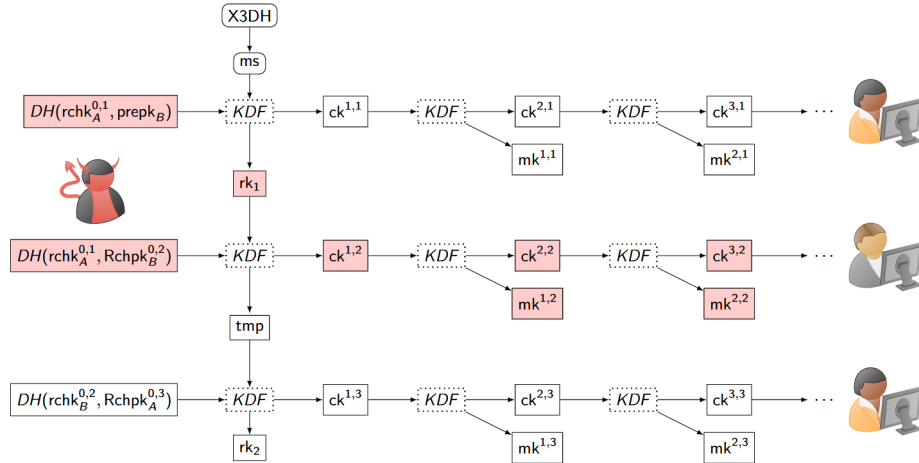
×

Invincibility lies in the defence, the possibility of victory in the attack

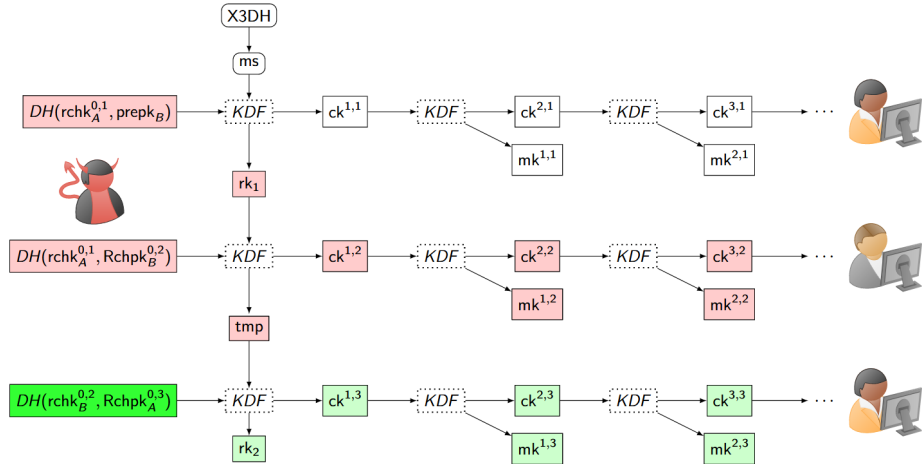


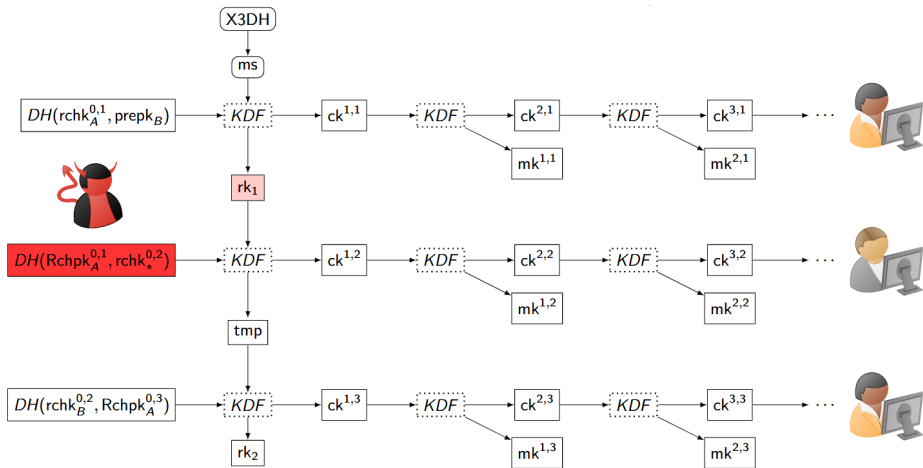


PASSIVE ATTACK ($\text{RCHK}_A^{0,1}, \text{RK}_1$)

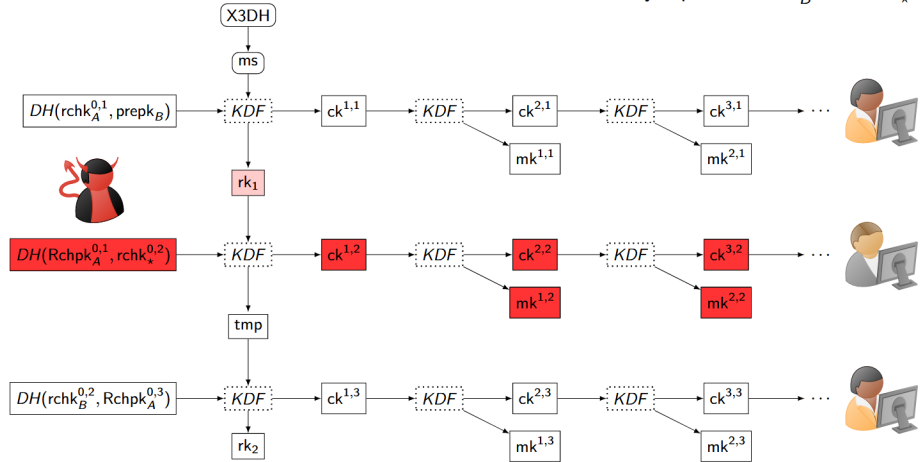


PASSIVE ATTACK ($\text{RCHK}_A^{0,1}, \text{RK}_1$)

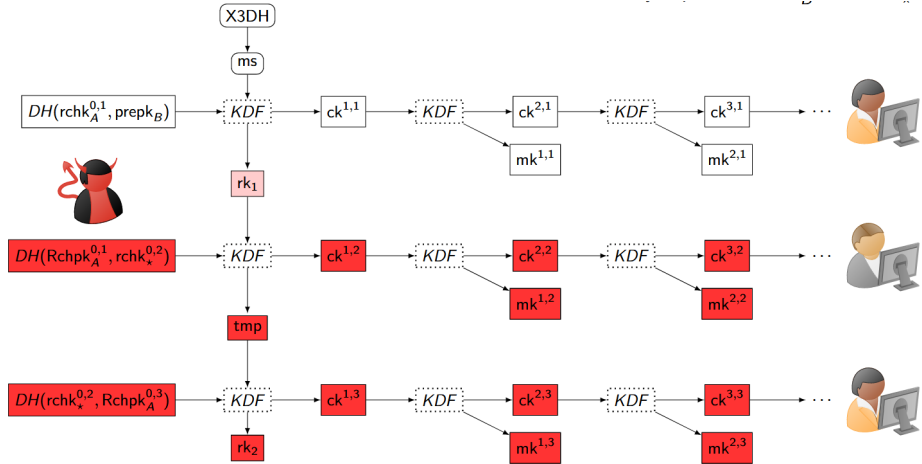




ACTIVE ATTACK (RK₁), NEW RCHK_{*}^{0,2}



ACTIVE ATTACK (RK₁), NEW RCHK_{*}^{0,2}



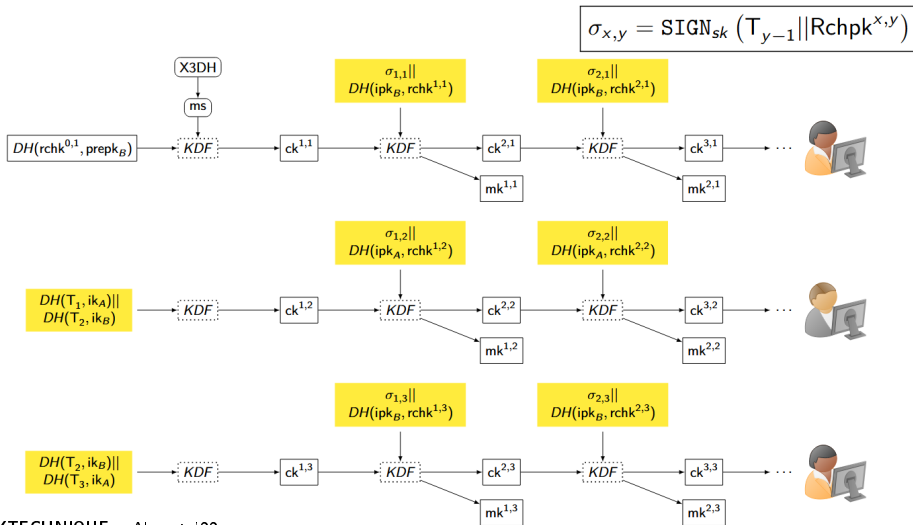


MARSHAL AND SAMURAI

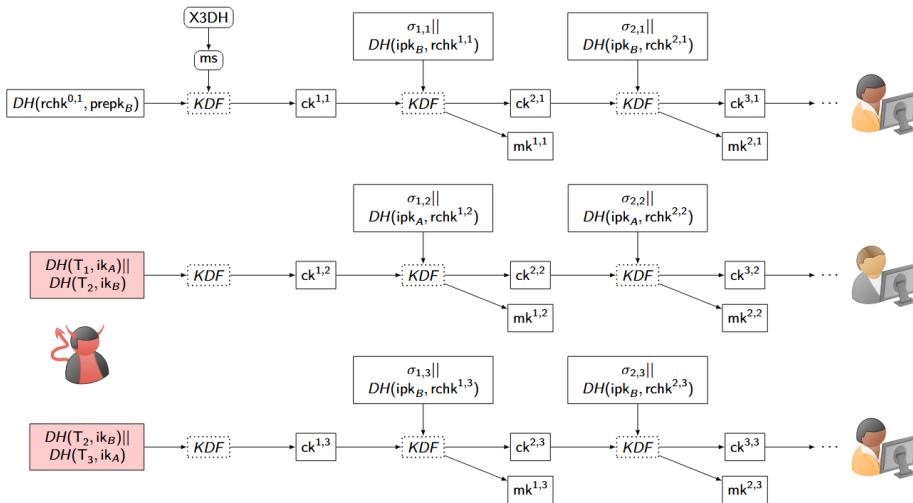
X



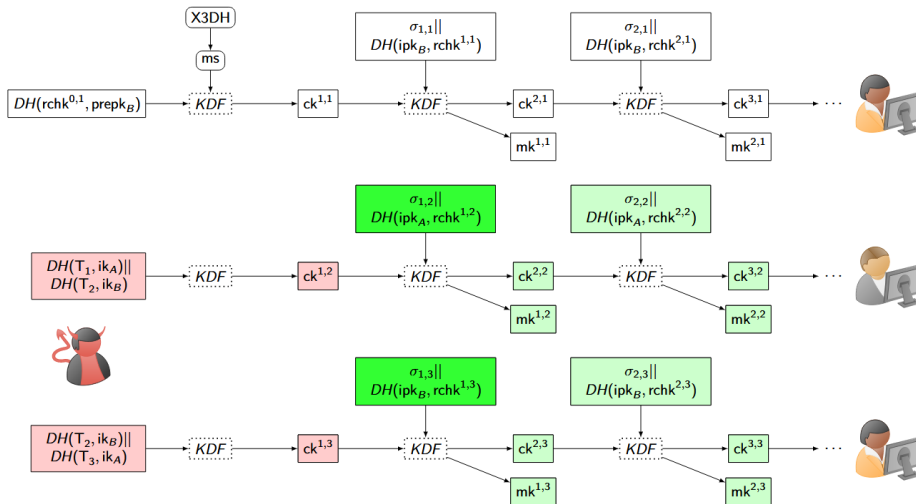
MARSHAL KEY SCHEDULE



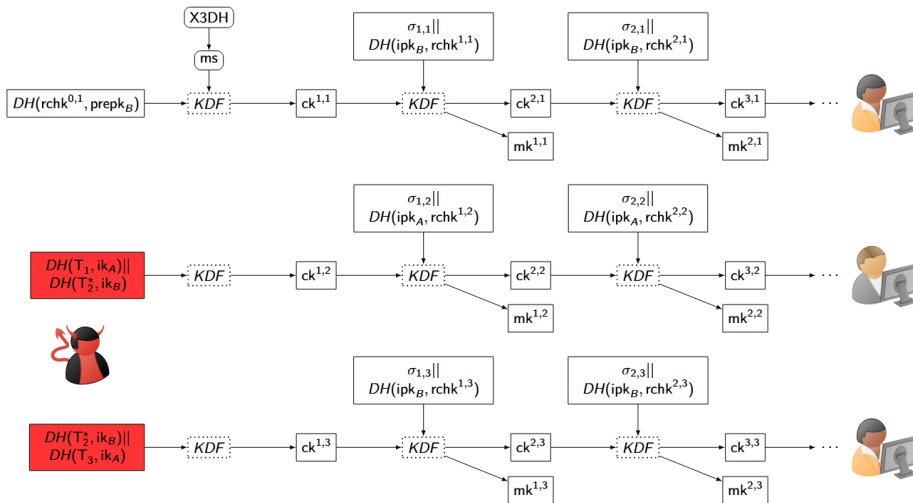
PASSIVE ATTACK (T_2)



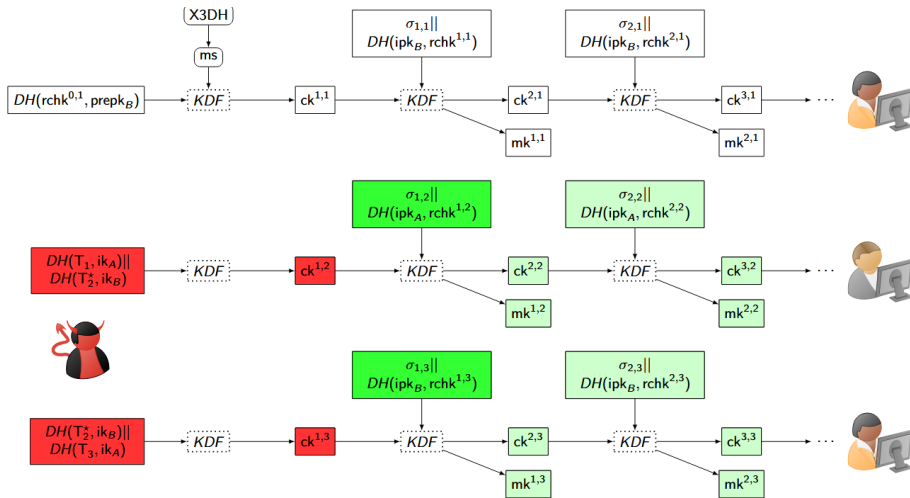
PASSIVE ATTACK (T_2)

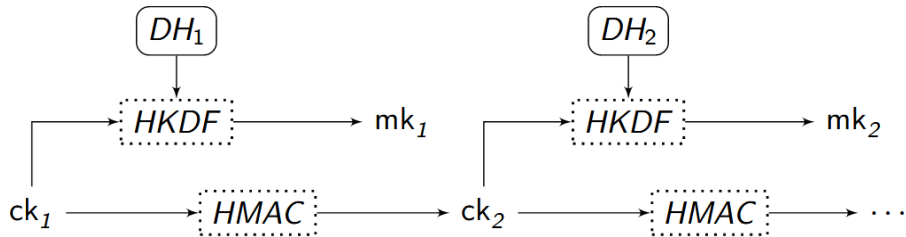


ACTIVE ATTACK (T_2^*)



ACTIVE ATTACK (T_2^*)





Supersedes the signature, by a MAC.

EFFICIENCY



	Signal	Marshal / Samurai
Session Setup	3.856ms	6.924ms
Message (1, y)	1.284ms	5.082ms
Message (100, y)	0.06ms	1.512ms

OUR MODEL TO QUANTIFY HEALING



GENERIC APPROACH



Security Properties

- Post Compromise Security
- Authentication
- ...

Protocols

- Signal
- Said
- Marshal

Adversaries

- Power
- Behavior
- ...

THREE CATEGORIES OF KEYS



Cross-session keys (long-term)

Keys that repeat in at least two sessions

Cross-stage keys (medium-term)

Keys that repeat in at least two stages of the same session, but do not repeat in two distinct sessions

Single-stage keys (ephemeral)

These are keys that only occur in one stage of one protocol instance



ADVERSARIES



Reach

- **Local adversaries:** can reveal single-stage keys
- **Medium adversaries:** can reveal single and cross-stage keys
- **Global adversaries:** can reveal all keys

Power

- **Active adversaries:** can send/receive messages → session hijacking
- **Passive adversaries:** try to learn as much as possible about future keys without actually interfering in the session too much

Access

- **Insider adversaries:** these attackers are the super-user (server)
- **Outsider adversaries:** these attackers cannot corrupt the server

ADVERSARIES



Reach

- **Local adversaries:** can reveal single-stage keys
- **Medium adversaries:** can reveal single and cross-stage keys
- **Global adversaries:** can reveal all keys

Power

- **Active adversaries:** can send/receive messages → **session hijacking**
- **Passive adversaries:** try to learn as much as possible about future keys without actually interfering in the session too much

Access

- **Insider adversaries:** these attackers are the super-user (server)
- **Outsider adversaries:** these attackers cannot corrupt the server

ADVERSARIES



Reach

- **Local adversaries:** can reveal single-stage keys
- **Medium adversaries:** can reveal single and cross-stage keys
- **Global adversaries:** can reveal all keys

Power

- **Active adversaries:** can send/receive messages → **session hijacking**
- **Passive adversaries:** try to learn as much as possible about future keys without actually interfering in the session too much

Access

- **Insider adversaries:** these attackers are the super-user (server)
- **Outsider adversaries:** these attackers cannot corrupt the server

POST COMPROMISE SECURITY QUANTIFICATION

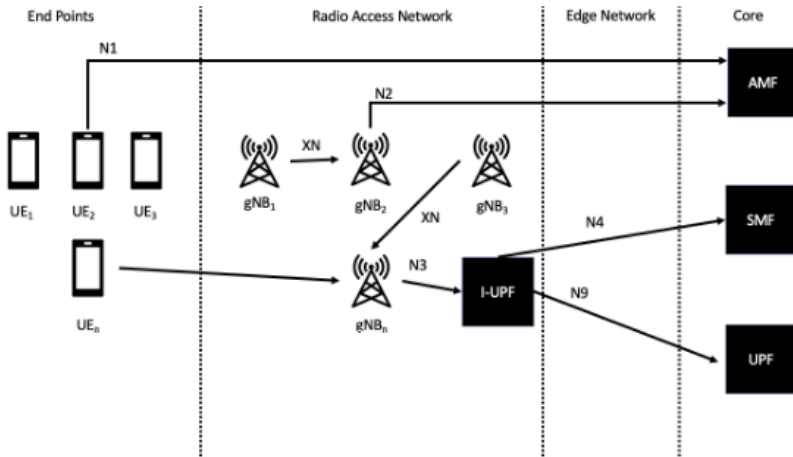


Outsider	Reach	Signal	SAID	Marshal
Passive	Global	$(\infty, 2)$	$(\infty, 2)$	$(1, 0)$
	Medium	$(\infty, 2)$	$(\infty, 2)$	$(1, 0)$
	Local	$(\infty, 1)$	$(1, 0)$	$(1, 0)$
Active	Global	(∞, ∞)	(∞, ∞)	(∞, ∞)
	Medium	(∞, ∞)	(∞, ∞)	$(1, 0)$
	Local	$(\infty, 1)$	$(1, 0)$	$(1, 0)$
Insider	Reach	Signal	SAID	Marshal
Passive	Global	$(\infty, 2)$	$(\infty, 2)$	$(1, 0)$
	Medium	$(\infty, 2)$	$(\infty, 2)$	$(1, 0)$
	Local	$(\infty, 1)$	$(\infty, 1)$	$(1, 0)$
Active	Global	(∞, ∞)	(∞, ∞)	(∞, ∞)
	Medium	(∞, ∞)	(∞, ∞)	(∞, ∞)
	Local	(∞, ∞)	(∞, ∞)	(∞, ∞)

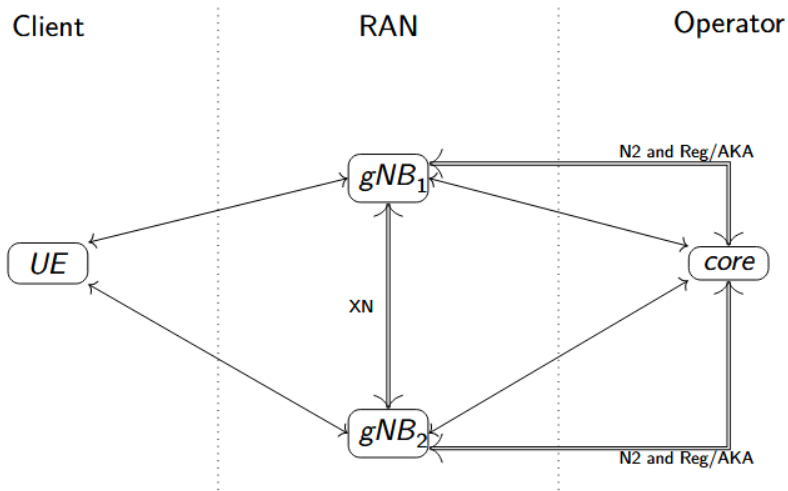
APPLICATION TO 5G



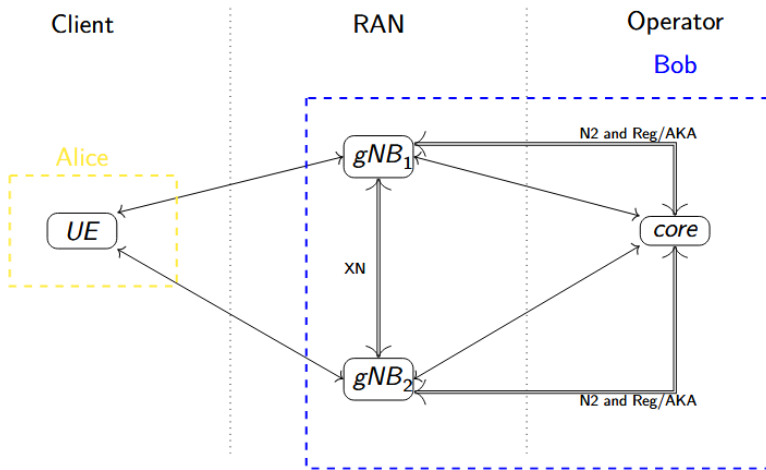
HANDOVER PROCEDURES IN 5G NETWORKS



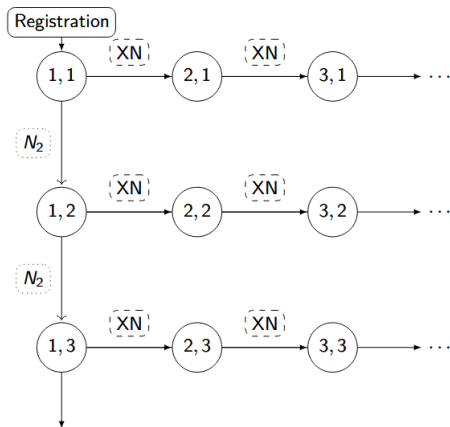
A FOCUS ON XN AND N2

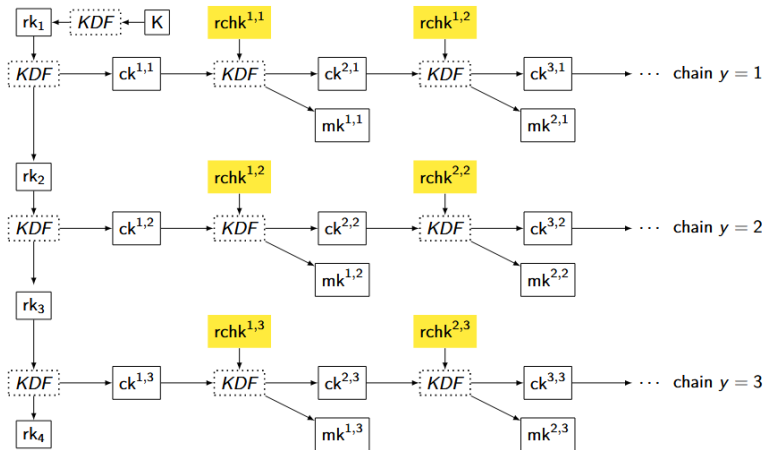


A FOCUS ON XN AND N2



COMING BACK TO OUR MODEL...





POST COMPROMISE SECURITY QUANTIFICATION



Outsider	Reach	5G	Improved 5G
Passive	Global	(∞, ∞)	$(1, 0)$
	Medium	(∞, ∞)	$(1, 0)$
	Local	$(\infty, 1)$	$(1, 0)$
Active	Global	(∞, ∞)	(∞, ∞)
	Medium	(∞, ∞)	(∞, ∞)
	Local	$(\infty, 1)$	$(\infty, 1)$
Insider	Reach	5G	Improved 5G
Passive	Global	(∞, ∞)	$(1, 0)$
	Medium	(∞, ∞)	$(1, 0)$
	Local	(∞, ∞)	$(1, 0)$
Active	Global	(∞, ∞)	(∞, ∞)
	Medium	(∞, ∞)	(∞, ∞)
	Local	(∞, ∞)	(∞, ∞)

WRAPPING UP

New Framework

- Taxonomy on adversaries/keys
- Healing speed
- Comparison between different protocols

Future Work

- Consider all features (out-of-order messages)
- Other protocols TLS, Matrix, Olvid, Wire
- Multi-party protocols / Groups (dynamic)
- Automated verification