Limits of Proofs: Social Foundations of Cryptography

Advanced Topics in ~~Cybersecurity~~Cryptography (7CCSMATC)

Martin R. Albrecht

Outline

The Indictment

Example: Local Adversaries

Example: Digital Signatures

Example: Forward Secrecy & Post-Compromise Security

The Indictment

"The Equivalent of a Written Signature" [DH76]

We note that the formulation of digital signatures also provides a clear statement of the essential ingredients of handwritten signatures. The ingredients are each person's ability to sign for him/herself, a universally agreed-upon verification procedure, and the belief (or assertion) that it is infeasible (or at least hard) to forge signatures in a manner that passes the verification procedure. It is not clear to what extent handwritten signatures do meet these requirements. In contrast, our treatment of digital-signature schemes provide precise statements concerning the extent to which digital signatures meet these requirements.[1]

[1] Oded Goldreich. Foundations of Cryptography: Basic Applications. Vol. 2. Cambridge, UK: Cambridge University Press, 2004. ISBN: 978-0511721656. DOI: 10.1017/CBO9780511721656, p.498.

Main Reference

Jean-François Blanchette. **Burdens of Proof: Cryptographic Culture and Evidence Law in the Age of Electronic Documents**. MIT Press, 2012, Chapter 4: The Equivalent of a Written Signature

"On these questions, the literature remains silent."

Goldreich has argued cryptographic practice proceeds through "the identification, conceptualization and rigorous definition of cryptographic tasks which capture natural security concerns." [...] How does one identify such "natural security concerns"? On these questions, the literature remains silent. Furthermore, neither of these approaches provide a rationale for the definitions of the "mutant" signatures described in the preceding section, which can hardly be accounted for on the basis of "natural requirements." — [Bla12, p.89]

"'Mutant' Signatures"

Goldreich has argued cryptographic practice proceeds through "the identification, conceptualization and rigorous definition of cryptographic tasks which capture natural security concerns." [...] How does one identify such "natural security concerns"? On these questions, the literature remains silent. *Furthermore, neither of these approaches provide a rationale for the definitions of the "mutant" signatures described in the preceding section, which can hardly be accounted for on the basis of "natural requirements." — [Bla12, p.89]*

Some Advanced Signature Notions

Group ... allow any member of the group can sign a document, but it is not possible to identify who did. If a dispute arises over the legitimacy of a signature, a group manager (who controls membership) can be called on to identify the signer.

Ring ... require no group managers and no prespecified group size.

Blind ... allow the signer affixes her signature to a document but she cannot later link the two together.

Designated-Verifier ... allow a receiver to convince himself that a message is not a forgery, but cannot transfer that conviction to a third party.

Motivation

These signature schemes represent fascinating new configurations of responsibility, liability, trust, and power within the signing process. In most cases, there are no obvious "real-world" equivalents to these mathematical constructs, and in most cases, it is difficult to imagine the specific context in which they might be applied. Nevertheless, it is standard practice for cryptographic papers to justify such schemes with a "motivation" narrative, a "real-world" scenario that aims to suggest a plausible practical application of the signature scheme. — [Bla12, p.82]

Example Motivation [RST01] I

To motivate the title for this paper, suppose that Bob (also known as "Deep Throat") is a member of the cabinet of Lower Kryptonia, and that Bob wishes to leak a juicy fact to a journalist about the escapades of the Prime Minister, in such a way that Bob remains anonymous, yet such that the journalist is convinced that the leak was indeed from a cabinet member.
[...]
A standard group signature scheme does not solve the problem, since it requires the prior cooperation of the other group members to set up, and leaves Bob vulnerable to later identification by the group manager, who may be controlled by the Prime Minister.
[...]

Example Motivation [RST01] II

The correct approach is for Bob to send the story to the journalist (through an anonymizer), signed with a ring signature scheme that names each cabinet member (including himself) as a ring member. The journalist can verify the ring signature on the message, and learn that it definitely came from a cabinet member. He can even post the ring signature in his paper or web page, to prove to his readers that the juicy story came from a reputable source. However, neither he nor his readers can determine the actual source of the leak, and thus the whistleblower has perfect protection even if the journalist is later forced by a judge to reveal his "source" (the signed document).

The Cryptographic Paper Genre

Like so many modular Lego pieces, cryptographic primitives and design patterns are assembled in new schemes and protocols exhibiting security properties with no obvious real-world equivalents. This creative process is one of the core professional activities of cryptographers, rewarded through conference presentations, journal publications, and commercial patents. Yet the cryptographic paper genre seems to require that these products of mathematical creativity be justified in some "real-world" setting, motivated either by their potential application, their evidential value, or the new threats they identify. These justificatory scenarios are remarkable in their assumptions that the properties of cryptographic objects, as designed and discussed by cryptographers, will translate transparently into the complex social settings they describe. — *[Bla12, p.84]*

Like so many modular Lego pieces, cryptographic primitives and design patterns are assembled in new schemes and protocols exhibiting security properties with no obvious real-world equivalents. This creative process is one of the core professional activities of cryptographers, rewarded through conference presentations, journal publications, and commercial patents. Yet the cryptographic paper genre seems to require that these products of mathematical creativity be justified in some "real-world" setting, motivated either by their potential application, their evidential value, or the new threats they identify. These justificatory scenarios are remarkable in their assumptions that the properties of cryptographic objects, as designed and discussed by cryptographers, will translate transparently into the complex social settings they describe. — [Bla12, p.84]

An Caveat on the Cryptographic Paper Genre

The claim is true for many but not all cryptographic papers, e.g. [GGSW13] starts with:

> *When we encrypt a message using a public-key encryption scheme, we allow the receiver to learn our message only if he knows a secret key corresponding to his public key. What if we don't really care if he knows a secret key, but we do care if he knows a solution to a crossword puzzle that we saw in the Times? Or if he knows a short proof for the Goldbach conjecture? Or, in general, the solution to some NP search problem? In this paper, we ask the question: Can we encrypt a message so that it can only be opened by a recipient who knows a witness to an NP relation?*

The Indictment

- The field of cryptography prides itself on its rigour, rightly so, as hopefully demonstrated in this module.
- Blanchette points out that this rigour is sorely lacking when it comes to the social relations that cryptography models.

It is our opinion that the design of cryptographic systems has to be based on firm foundations; whereas ad-hoc approaches and heuristics are a very dangerous way to go.[a]

[a] Oded Goldreich. On the Foundations of Modern Cryptography (Invited Lecture). In: *CRYPTO'97*. Ed. by Burton S. Kaliski Jr. Vol. 1294. LNCS. Springer, Berlin, Heidelberg, Aug. 1997, pp. 46–74. DOI: 10.1007/BFb0052227.

Proposition

Cryptography is a social science unaware of itself.[2]

[2] Less catchy, more complete: cryptography is mathematics, complexity theory, engineering and social science.

Two Clarifications

Studying the social foundations of cryptography with same rigour as we study the complexity-theoretic or mathematical foundations of cryptography …

- is not an area of **human-computer interaction** (HCI) such as usable security.
- There is no computer to interact with when we ask “what is a digital signature scheme?”

- is not an area of **applied cryptography**, which applies cryptographic solutions to real or presumed-real (see “cryptographic paper genre”) settings.
- Sciences tend to a have an “applied” department, but while Alice and Bob do not live in physics or mathematics paper introductions, they do frequently appear in cryptography.
- The definition of a ring signature and its security properties is a definitional task, not an application of cryptography.

Example: Local Adversaries

Local Adversaries

Ksenia Ermoshina, Harry Halpin, and Francesca Musiani. Can johnny build a protocol? co-ordinating developer and user intentions for privacy-enhanced secure messaging protocols. In: *European Workshop on Usable Security*. 2017

“Currently developers simply imagine what properties users likely need”

Currently developers simply imagine what properties users likely need, and these properties may or may not actually satisfy the needs of end-users. In particular, high-risk users may care about very different properties than low-risk users in terms of their threat models. If developers themselves are relatively low-risk users and building tools aimed at high-risk users, then the tools may or may not match the needs of these high-risk users.

“DEVELOPERS ARE MOTIVATED BY HIGH-RISK ACTIVISTS, BUT HAVE LITTLE ACTUAL CONTACT”

Developer motivation was quite wide-ranging, but largely could be divided between those who wanted to start privacy-enhanced businesses that would serve both low and high-risk users to those who were primarily motivated by protecting high-risk users due to human rights concerns that are more traditionally dealt with by the NGO sector. [...] Strangely, it appears that developers are motivated by high-risk activists, but have little actual contact with high-risk users in their systems.

As has been observed among our interviews, in more high-risk situations such as Ukraine, the choice of secure messaging application can be due to the politics of its country of origin. These high-risk activists exclude applications and online services that have servers on the territory of Russian Federation or Ukraine and prefer American-based services, with even trainers advocating usages of Gmail and Facebook. Similar dynamics were observed in Iran (with no adoption of GPG and strong preference for Gmail with two-factor authentication), and Egypt (where WhatsApp is popular as the United States is considered as not being part of the threat model).

Trusted Third Parties II

'The most important thing for us is to convince people to stop using Russian services like mail.ru or yandex.ru. It is a direct backdoor to FSB office. We recommend also to switch from Telegram to WhatsApp. We recommend Gmail with two-factor authentication over PGP. It is easier to explain and people are already used to the interface [...] I don't think US will give out data on Ukrainians to Russians. [...] However, after Trump everything may change' (V., trainer, Ukraine)."

Differing Adversaries

High-risk users defined their threat model against a local active adversary, often the police or secret agencies of their government or a nearby hostile government, rather than a global passive adversary such as the NSA. In contrast, developers usually view their threat model as the NSA, a global powerful adversary, despite the lack of attention to privacy properties like metadata collection in secure messaging protocols.

However, high-risk users are not homogeneous, as the social and geopolitical differences between high-risk users lead to vastly different eco-systems of applications.

Differing Adversaries

High-risk users defined their threat model against a local active adversary, often the police or secret agencies of their government or a nearby hostile government, rather than a global passive adversary such as the NSA. In contrast, developers usually view their threat model as the NSA, a global powerful adversary, despite the lack of attention to privacy properties like metadata collection in secure messaging protocols.

However, high-risk users are not homogeneous, as the social and geopolitical differences between high-risk users lead to vastly different eco-systems of applications.

Differing Adversaries

High-risk users defined their threat model against a local active adversary, often the police or secret agencies of their government or a nearby hostile government, rather than a global passive adversary such as the NSA. In contrast, developers usually view their threat model as the NSA, a global powerful adversary, despite the lack of attention to privacy properties like metadata collection in secure messaging protocols.

However, high-risk users are not homogeneous, as the social and geopolitical differences between high-risk users lead to vastly different eco-systems of applications.

Example: Digital Signatures

Disputes Between Transmitter and Receiver

Authentication is at the heart of any system involving contracts and billing. Without it, business cannot function. Current electronic authentication systems cannot meet the need for a purely digital, unforgeable, message dependent signature. They provide protection against third party forgeries, but do not protect against disputes between transmitter and receiver. — [DH76]

If electronic mail systems are to replace the existing paper mail system for business transactions, "signing" an electronic message must be possible. The recipient of a signed message has proof that the message originated from the sender. This quality is stronger than mere authentication [...]; the recipient can convince a "judge" that the signer sent the message. To do so, he must convince the judge that he did not forge the signed message himself! — [RSA78]

Non-Repudiation

- [DH76] and [RSA78] were both concerned with what is called “non-repudiation” in law.
- They envisioned contracts and the disputes arising from them as a major area of contention where cryptography would regulate behaviour.
- Their vision did not pan out:
  1. The standard notion of security for a signature scheme is EUF-CMA which does not capture “non-repudiation”
  2. E-commerce took off without cryptographic digital signatures under contracts being common.

EUF-CMA

4.B.2 Security Definition for Digital Signatures NIST intends to standardize one or more schemes that enable existentially unforgeable digital signatures with respect to an adaptive chosen message attack. (This property is generally denoted *EUF-CMA security* in academic literature.)

The above security definition should be taken as a statement of what NIST will consider to be a relevant attack. Submitted algorithms for digital signatures will be evaluated based on how well they appear to provide this property when used as specified by the submitter. Submitters are not required to provide a proof of security, although such proofs will be considered if they are available.

For the purpose of estimating security strengths, it may be assumed that the attacker has access to signatures for no more than 2^{64} chosen messages; however, attacks involving more messages may also be considered. Additionally, it should be noted that NIST is primarily concerned with attacks that use classical (rather than quantum) queries to the signing oracle.

https://csrc.nist.gov/csrc/media/Projects/pqc-dig-sig/documents/call-for-proposals-dig-sig-sept-2022.pdf

Recap: EUF-CMA

EUF-CMA	S(m)
1 : $\mathcal{Q} \leftarrow \emptyset$;	1 : $\sigma \leftarrow \Sigma.\mathsf{Sign}(\mathsf{sk}, m)$
2 : $\mathsf{vk}, \mathsf{sk} \leftarrow\!\!\$\ \Sigma.\mathsf{KeyGen}(1^\lambda)$;	2 : $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{(m, \sigma)\}$
3 : $(m^\star, \sigma^\star) \leftarrow\!\!\$\ \mathcal{A}^{\mathsf{S}}(\mathsf{vk})$;	3 : **return** σ
4 : **return** $(m^\star, \cdot) \notin \mathcal{Q} \wedge \Sigma.\mathsf{Verify}(\mathsf{vk}, \sigma^\star, m^\star) = 1$	

$$\mathsf{Adv}^{\text{euf-cma}}_{\mathcal{A},\Sigma}(\lambda) := \Pr[\text{EUF-CMA}^{\mathcal{A}}_{\Sigma}(\lambda) \Rightarrow 1]$$

Malicious-Strong Universal Exclusive Ownership

M-S-UEO

1 : $m_0, m_1, \sigma, \mathsf{vk}_0, \mathsf{vk}_1 \leftarrow\!\$\ \mathcal{A}()$

2 : $b_0 \leftarrow \Sigma.\mathsf{Verify}(\mathsf{vk}_0, \sigma, m_0)$

3 : $b_1 \leftarrow \Sigma.\mathsf{Verify}(\mathsf{vk}_1, \sigma, m_1)$

4 : **return** $b_0 \wedge b_1 \wedge \mathsf{vk}_0 \neq \mathsf{vk}_1$

$$\mathsf{Adv}^{\text{m-s-ueo}}_{\mathcal{A},\Sigma}(\lambda) \coloneqq \Pr[\text{M-S-UEO}^{\mathcal{A}}_{\Sigma}(\lambda) \Rightarrow 1]$$

EUF-CMA $\not\Rightarrow$ M-S-UEO

- We have seen (in the “rewinding” lecture) that Schnorr signatures are EUF-CMA in the ROM and if discrete logarithms are hard.
- Below we give an attack breaking M-S-EUO.

Recap: Schnorr

Let $H : \mathbb{G} \times \{0,1\}^* \to \mathbb{Z}_p$ be a hash function

Gen

$\mathsf{sk} \coloneqq x \leftarrow\!\$\, \mathbb{Z}_p; \quad \mathsf{vk} \coloneqq X \coloneqq G^x$

Claus-Peter Schnorr. Efficient Identification and Signatures for Smart Cards. In: *CRYPTO'89*. Ed. by Gilles Brassard. Vol. 435. LNCS. Springer, New York, Aug. 1990, pp. 239–252. DOI: 10.1007/0-387-34805-0_22

Sign(sk, m)

1. $y \leftarrow\!\$\, \mathbb{Z}_p$ and set $Y \leftarrow G^y$
2. $c \leftarrow H(Y, m)$
3. $z \leftarrow y - c \cdot x$

$\sigma \coloneqq (Y, z)$

Verify(vk, σ, m)

1. $c \leftarrow H(Y, m)$
2. $Y \stackrel{?}{=} G^z \cdot X^c = G^z \cdot G^{c \cdot x} = G^{y - c \cdot x + c \cdot x}$

M-S-UEO Attack

Attack:

1. Sample $x_0 \leftarrow\!\$\ \mathbb{Z}_p$ and set $\mathsf{vk}_0 := X_0 := G^{x_0}$
2. Pick m_0, m_1.
3. Sample $y \leftarrow\!\$\ \mathbb{Z}_p$ and set $Y \leftarrow G^y$.
4. Compute $c_0 \leftarrow H(Y, m_0)$ and $z \leftarrow y - c_0 \cdot x_0$
5. Set $\sigma := (Y, z)$.
6. $c_1 \leftarrow H(Y, m_1)$
7. Set $x_1 := c_0/c_1 \cdot x \bmod \mathbb{Z}_p$ and $\mathsf{vk}_1 := X_1 := G^{x_1}$
8. Output $(m_0, m_1, \sigma, \mathsf{vk}_0, \mathsf{vk}_1)$

Verification:

$$
\begin{aligned}
c_0 &\leftarrow H(Y, m_0) \\
c_1 &\leftarrow H(Y, m_1) \\
Y &= G^z \cdot X_0^{c_0} = G^z \cdot G^{c_0 \cdot x_0} \\
&= G^{y - c_0 \cdot x_0 + c_0 \cdot x_0} \\
Y &= G^z \cdot X_1^{c_1} = G^z \cdot G^{c_1 \cdot x_1} \\
&= G^{y - c_0 \cdot x_0 + c_1 \cdot x_1} \\
&= G^{y - c_0 \cdot x_0 + c_1 \cdot c_0 / c_1 \cdot x_0} \\
&= G^{y - c_0 \cdot x_0 + c_0 \cdot x_0}
\end{aligned}
$$

Easy Fix [CDFFJ21]

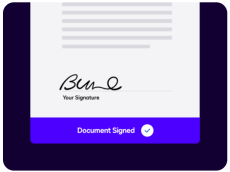
$$c \leftarrow H(Y, m, X)$$

Summary

- A scheme that does not satisfy M-S-UEO cannot provide non-repudiation
  - "I, owning X_1, did not sign m_1, but someone, owning X_2 did sign m_2"
- EUF-CMA does not imply M-S-UEO
- Digital signatures – by default – do not provide the guarantees envisioned by their inventors.
- They wanted these guarantees for electronic commerce, but somehow economic commerce flourishes without cryptographic digital signatures being standard for signing contracts.

“The Equivalent of a Written Signature” [DH76]

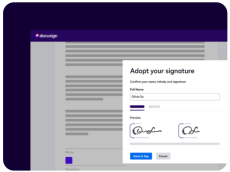
Docusign supports all levels of signatures defined by eIDAS



Electronic Signature

Docusign eSignature is trusted by hundreds of millions of users world-wide and meets electronic signature regulations, including eIDAS.

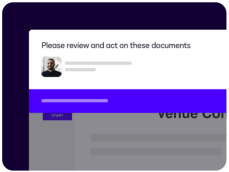
Learn More ›



Advanced Electronic Signature

Add Docusign ID Verification to your electronic signature process to validate signer identity and support AES. This is needed in use cases that require stricter identity assurance.

Learn More ›



Qualified Electronic Signatures

A QES is the legal equivalent of a written signature and has special legal status in the EU. Meet EU and UK QES standards by combining eSignature with our ID Verification for EU Qualified offering or partner solutions.

Learn More ›

“The Equivalent of a Written Signature” [DH76]

Although now firmly established in the security literature, non-repudiation never fully gained traction as a legal concept, as it seemed to preempt the very process by which the qualities of a given form of evidence are gradually established through the adversarial process. — [Bla12, p.11]

Deniability

- In some applications, the opposite of non-repudiation may be desirable: deniability.
- Deniability is the guarantee that I can deny having sent a certain message after the fact.
- It may be desirable in a one-to-one chat to prevent Bob from taking Alice' messages and present these to Charley to convince him of the mean things Alice has said about him.
- Not being able to do that is closer to a face to face conversation than being able to.
- Like with "non-repudiation", the law may not care much about "deniability" at the cryptographic level

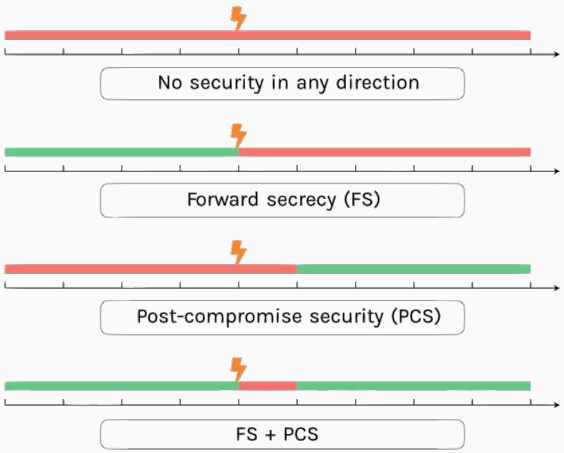
Reading Recommendation

Matthew Green. *Ok Google: please publish your DKIM secret keys.* 2020
`https://blog.cryptographyengineering.com/2020/11/16/`

Example: Forward Secrecy & Post-Compromise Security

Martin R. Albrecht, Jorge Blasco, Rikke Bjerg Jensen, and Lenka Mareková. **Collective Information Security in Large-Scale Urban Protests: the Case of Hong Kong.** In: *USENIX Security 2021*. Ed. by Michael Bailey and Rachel Greenstadt. USENIX Association, Aug. 2021, pp. 3363–3380

Forward Secrecy & Post-Compromise Security



https://pqshield.com/post-quantum-messaging-examining-apples-new-pq3-protocol/

- **Forward Secrecy** (FS) [Gün90; Kra05] means the protection of past messages in the event of a later compromise of an involved party
- **Post-Compromise Security** (PCS) [CCG16; CHK19] means the protection of future messages some time after a compromise.

On Post-Compromise Security [CCG16]

[M]odern secure systems are designed assuming that some compromise might eventually occur, and aim instead to limit its scope. For example, devices frequently fall temporarily under adversarial control: later-removed malware, short-term physical access, and confiscation at a border crossing all describe scenarios where a potential adversary has access to a device which is then returned to the original owner. In the traditional paradigm, we cannot make security guarantees about these post-compromise cases.

On Post-Compromise Security [CCG16]

[M]odern secure systems are designed assuming that some compromise might eventually occur, and aim instead to limit its scope. For example, devices frequently fall temporarily under adversarial control: later-removed malware, short-term physical access, and confiscation at a border crossing all describe scenarios where a potential adversary has access to a device which is then returned to the original owner. In the traditional paradigm, we cannot make security guarantees about these post-compromise cases.

Recap: Public-Key Encryption (PKE)

A Public-Key Encryption (PKE) scheme is a triple of PPT algorithms $(\mathsf{KeyGen}, \mathsf{Enc}, \mathsf{Dec})$ with the following syntax and operation:

KeyGen The key generation algorithm is a PPT algorithm taking as input a security parameter 1^λ and outputs a public/secret key-pair $(\mathsf{pk}, \mathsf{sk})$.

Enc The encryption algorithm is a PPT algorithm taking as input a public-key pk and a message m and outputs an encryption of m under pk.

Dec The decryption algorithm is a deterministic algorithm taking as input a ciphertext c and a secret-key sk, and outputs a message m (or an error message $\perp$ indicating a decryption failure).

Recap: IND-CPA

$\mathsf{IND\text{-}CPA}_{\mathsf{PKE}}$	$\mathsf{C}(m_0, m_1)$
$\mathsf{pk}, \mathsf{sk} \leftarrow\!\!\$\ \mathsf{KeyGen}(1^\lambda)$	**assert** $\lvert m_0\rvert = \lvert m_1\rvert$
$b \leftarrow\!\!\$\ \{0,1\}$	$c \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{pk}, m_b)$
$b' \leftarrow \mathcal{D}^{\mathsf{C}}(\mathsf{pk})$	**return** c
return $b = b'$	

$$\mathsf{Adv}^{\text{ind-cpa}}_{PKE}(\mathcal{D}) = |\Pr[\mathsf{IND\text{-}CPA}^{\mathcal{D}} = 1] - 1/2|$$

Key-Evolving Public-Key Encryption (KE-PKE) [CHK07]

A Key-Evolving Public-Key Encryption (PKE) scheme is a four-tuple of PPT algorithms (KeyGen, Upd, Enc, Dec) with the following syntax and operation:

KeyGen The key generation algorithm is a PPT algorithm taking as input a security parameter 1^λ and a total number of time periods N and outputs a public key pk and an initial secret key sk_0.

Upd The update algorithm is a deterministic algorithm taking as input a public-key pk, an index $i < N$ of the current time period and the associated secret key sk_i and outputs a secret key sk_{i+1} for the following time period.

Enc The encryption algorithm is a PPT algorithm taking as input a public-key pk, an index $i < N$ of the current time period and a message m and outputs an encryption of m under pk.

Dec The decryption algorithm is a deterministic algorithm taking as input a ciphertext c, an index $i < N$ of the current time and an associated secret-key sk_i, and outputs a message m (or an error message $\perp$ indicating a decryption failure).

FS-IND-CPA [CHK07]

FS-IND-CPA$_{\mathsf{PKE}}$

1 : $\mathsf{pk}, \mathsf{sk}_0 \leftarrow\!\!\$\ \mathsf{KeyGen}(1^\lambda)$
2 : $b \leftarrow\!\!\$\ \{0,1\}$
3 : $\mathsf{chal}, \mathsf{corr} \leftarrow \bot, \bot$
4 : $b' \leftarrow \mathcal{D}^{\mathrm{C},\mathrm{X}}(\mathsf{pk})$
5 : $\mathsf{valid} \leftarrow \mathsf{corr} > \mathsf{chal}$
6 : **return** $b = b' \wedge \mathsf{valid}$

C(i, m_0, m_1)

1 : **assert** $|m_0| = |m_1|$
2 : **assert** $\mathsf{chal} = \bot$
3 : **assert** $i \in \{0 \dots N-1\}$
4 : $c \leftarrow\!\!\$\ \mathsf{Enc}(\mathsf{pk}, i, m_b)$
5 : $\mathsf{chal} \leftarrow i$
6 : **return** c

X(i)

1 : **assert** $\mathsf{corr} = \bot$
2 : **assert** $i \in \{0 \dots N-1\}$
3 : **for** $0 \le j < i$
4 : $\quad \mathsf{sk}_{j+1} \leftarrow\!\!\$\ \mathsf{Upd}(\mathsf{pk}, j, \mathsf{sk}_j)$
5 : $\mathsf{corr} \leftarrow i$
6 : **return** sk_i

On FS-IND-CPA

- FS-IND-CPA is difficult to achieve, meaning we need “heavy machinery” for it
  - Look up: “Hierarchical Identity-Based Encryption”, “Puncturable Encryption” and “Bloom Filter Encryption” [GM15; DJSS18][3]
- I am giving the FS-IND-CPA definition because it is conceptually easy
- In a messaging setting, achieving forward secrecy is much easier, but the security notions for authenticated key exchange that are at play here are much more complex
  - This is because they model interactions between many parties (Alice, Bob, Charley, Dave, ...)
  - We will see a high-level example shortly

[3] Actually, look up “bloom filters” if you do not know what those are, they are very cool!

Yet Another Key-Evolving Public-Key Encryption (I made it up!)

KeyGen The key generation algorithm is a PPT algorithm taking as input a security parameter 1^λ and outputs a public key pk and an initial secret key sk_0.

Upd The update algorithm is a deterministic algorithm taking as input a public-key pk_i, a secret key sk_i and some randomness r and outputs a public/secret key pair $(\mathsf{pk}_{i+1}, \mathsf{sk}_{i+1})$ for the following time period.

Enc The encryption algorithm is a PPT algorithm taking as input a public-key pk_i and a message m and outputs an encryption of m under pk_i.

Dec The decryption algorithm is a deterministic algorithm taking as input a ciphertext c and a secret-key sk_i, and outputs a message m (or an error message $\perp$ indicating a decryption failure).

Post-Compromise Security

PCS-IND-CPA$_{\text{PKE}}$

1: $\text{pk}_0, \text{sk}_0 \leftarrow\$ \text{KeyGen}(1^\lambda)$
2: $i \leftarrow 0$; $\text{corr} \leftarrow \bot$
3: $b \leftarrow\$ \{0,1\}$
4: $b' \leftarrow \mathcal{D}^{\text{C},\text{X},\text{U}}(\text{pk}_0)$
5: **return** $b = b'$

C(m_0, m_1)

1: **assert** $|m_0| = |m_1|$
2: **assert** $\text{corr} \neq \bot$
3: **assert** $\text{corr} < i$
4: $c \leftarrow\$ \text{Enc}(\text{pk}, m_b)$
5: **return** c

U()

1: $r \leftarrow\$ \{0,1\}^\lambda$
2: $\text{pk}_{i+1}, \text{sk}_{i+1} \leftarrow \text{Upd}(pk_i, sk_i, r)$
3: $i \leftarrow i + 1$
4: **return** pk_{i+1}

X()

1: **assert** $\text{corr} = \bot$
2: $\text{corr} \leftarrow i$
3: **return** sk_i

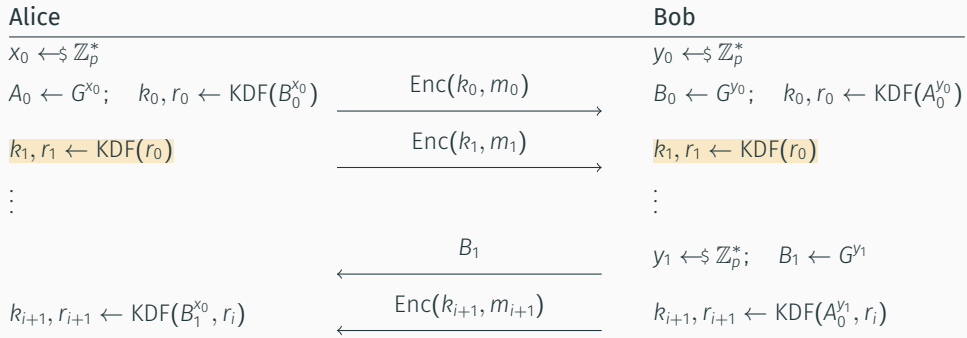
Signal's Double Ratchet: Rough Idea

Diffie-Hellman key exchanges to establish keys for symmetric IND-CCA secure encryption:

Alice		Bob
$x_0 \leftarrow\!\$\ \mathbb{Z}_p^*$		$y_0 \leftarrow\!\$\ \mathbb{Z}_p^*$
$A_0 \leftarrow G^{x_0}$; $\quad k_0, r_0 \leftarrow \mathrm{KDF}(B_0^{x_0})$	$\xrightarrow{\mathrm{Enc}(k_0, m_0)}$	$B_0 \leftarrow G^{y_0}$; $\quad k_0, r_0 \leftarrow \mathrm{KDF}(A_0^{y_0})$
$k_1, r_1 \leftarrow \mathrm{KDF}(r_0)$	$\xrightarrow{\mathrm{Enc}(k_1, m_1)}$	$k_1, r_1 \leftarrow \mathrm{KDF}(r_0)$
$\vdots$		$\vdots$
	$\xleftarrow{B_1}$	$y_1 \leftarrow\!\$\ \mathbb{Z}_p^*$; $\quad B_1 \leftarrow G^{y_1}$
$k_{i+1}, r_{i+1} \leftarrow \mathrm{KDF}(B_1^{x_0}, r_i)$	$\xleftarrow{\mathrm{Enc}(k_{i+1}, m_{i+1})}$	$k_{i+1}, r_{i+1} \leftarrow \mathrm{KDF}(A_0^{y_1}, r_i)$

FS & PCS in the Signal Protocol

Signal's "double ratchet" exists to achieve forward secrecy and post-compromise security



When a party sends a message, they will "ratchet forward" their symmetric secret key using a key derivation function to achieve **forward-secrecy**.

Key Derivation Functions

A key derivation function is a pseudorandom generator (PRG), see first lecture, that handles non-uniform inputs.

FS & PCS in the Signal Protocol

Signal's "double ratchet" exists to achieve forward secrecy and post-compromise security

Alice		Bob
$x_0 \leftarrow\$ \mathbb{Z}_p^*$		$y_0 \leftarrow\$ \mathbb{Z}_p^*$
$A_0 \leftarrow G^{x_0}$; $k_0, r_0 \leftarrow \mathrm{KDF}(B_0^{x_0})$	$\mathrm{Enc}(k_0, m_0)$ →	$B_0 \leftarrow G^{y_0}$; $k_0, r_0 \leftarrow \mathrm{KDF}(A_0^{y_0})$
$k_1, r_1 \leftarrow \mathrm{KDF}(r_0)$	$\mathrm{Enc}(k_1, m_1)$ →	$k_1, r_1 \leftarrow \mathrm{KDF}(r_0)$
⋮		⋮
	← B_1	$y_1 \leftarrow\$ \mathbb{Z}_p^*$; $B_1 \leftarrow G^{y_1}$
$k_{i+1}, r_{i+1} \leftarrow \mathrm{KDF}(B_1^{x_0}, r_i)$	← $\mathrm{Enc}(k_{i+1}, m_{i+1})$	$k_{i+1}, r_{i+1} \leftarrow \mathrm{KDF}(A_0^{y_1}, r_i)$

When a party responds, they run a new Diffie-Hellman key exchange and add the output to the KDF to achieve **post-compromise security**

PCS: Passive Flavour

Post-compromise security protects against an adversary that becomes passive at some point after a compromise.[4] The notion is otherwise unachievable.

- An adversary that has all secrets of Alice and actively controls the network can simulate Alice to Bob.
- The adversary can perform all steps Alice can perform.

[4] Passive: does not interfere with the protocol, it only observes.

No PCS in Signal Chats

The PCS guarantee is for Signal sessions between Alice and Bob (roughly the chain of r_i)

- Signal, the app, supports opening several such sessions in parallel.
- As a consequence it does not provide these post-compromise security guarantees for chats: an adversary can start a new session to impersonate Alice to Bob using the keys it compromised.
  - Above, this would mean starting again from A_0.
- This does not violate the security proof because of how post-compromise security is defined.
- I am not aware of a used-in-practice messenger that actually provides PCS for chats.

Cas Cremers, Charlie Jacomme, and Aurora Naska. **Formal Analysis of Session-Handling in Secure Messaging: Lifting Security from Sessions to Conversations.** In: *USENIX Security 2023.* Ed. by Joseph A. Calandrino and Carmela Troncoso. USENIX Association, Aug. 2023, pp. 1235–1252

FS & PCS in Anti-ELAB Protests in Hong Kong [ABJM21]

Abstract: *The Anti-Extradition Law Amendment Bill protests in Hong Kong present a rich context for exploring information security practices among protesters due to their large-scale urban setting and highly digitalised nature. We conducted in-depth, semi-structured interviews with 11 participants of these protests. Research findings reveal how protesters [...] developed a variety of strategies to detect compromises and to achieve forms of forward secrecy and post-compromise security when group members were (presumed) arrested. [...]*

- The compromise the participants were most concerned about was during and after an arrest.[a]
- Here, they were concerned with both forward secrecy and post-compromise security.

[a] Most similar to "confiscation at a border crossing" in [CCG16].

FS & PCS in Anti-ELAB Protests in Hong Kong [ABJM21]

Protest participants attempted to detect when members of their affinity groups were arrested. In that event, they would aim to achieve:

Forward Secrecy by remotely deleting messages on the arrested person's phone; and

Post-Compromise Security by removing the arrested person from their group chats.

FS & PCS Notions Differed from those in the Literature I

1. A cryptographic scheme achieving forward secrecy would not achieve the notion of forward secrecy desired by the participants in the study as messages remained stored on the recipient's device.
   - That is, the participants assumed and aimed to protect against a compromise that reveals not only key material but also the entire chat history (stored on the phone).
2. A security goal of the participants in the study was to protect themselves during the compromise not just afterwards.
   - There were a variety of tactics attempting to detect and control compromise as it happens, including location monitoring, timed messages and revocation of administrator capabilities, all done on behalf of the compromised person by the remaining group members.
3. The notion of post-compromise security was at a group level (removing the compromised party) rather than for the compromised party.

FS & PCS Notions Differed from those in the Literature II

Incomparable

The adversary model of the participants in the study was both stronger (the adversary also compromises the chat history; protection against an adversary during a compromise is intended) and weaker (detectable) than those in the literature, i.e. the resulting security notions are incomparable.

Fin

"While the immediate objects of cryptography are not social relations, it presumes and models them." — https://social-foundations-of-cryptography.gitlab.io/about

References I

[ABJM21] Martin R. Albrecht, Jorge Blasco, Rikke Bjerg Jensen, and Lenka Mareková. Collective Information Security in Large-Scale Urban Protests: the Case of Hong Kong. In: *USENIX Security 2021*. Ed. by Michael Bailey and Rachel Greenstadt. USENIX Association, Aug. 2021, pp. 3363–3380.

[Bla12] Jean-François Blanchette. Burdens of Proof: Cryptographic Culture and Evidence Law in the Age of Electronic Documents. MIT Press, 2012.

[CCG16] Katriel Cohn-Gordon, Cas J. F. Cremers, and Luke Garratt. On Post-compromise Security. In: *IEEE 29th Computer Security Foundations Symposium, CSF 2016, Lisbon, Portugal, June 27 - July 1, 2016*. IEEE Computer Society, 2016, pp. 164–178.

References II

[CDFFJ21] Cas Cremers, Samed Düzlü, Rune Fiedler, Marc Fischlin, and Christian Janson. **BUFFing signature schemes beyond unforgeability and the case of post-quantum signatures.** In: *2021 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2021, pp. 1696–1714. DOI: 10.1109/SP40001.2021.00093.

[CHK07] Ran Canetti, Shai Halevi, and Jonathan Katz. **A Forward-Secure Public-Key Encryption Scheme.** In: *Journal of Cryptology* 20.3 (July 2007), pp. 265–294. DOI: 10.1007/s00145-006-0442-5.

[CHK19] Cas Cremers, Britta Hale, and Konrad Kohbrok. **Efficient Post-Compromise Security Beyond One Group.** Cryptology ePrint Archive, Report 2019/477. 2019. URL: https://eprint.iacr.org/2019/477.

References III

[CJN23] Cas Cremers, Charlie Jacomme, and Aurora Naska. **Formal Analysis of Session-Handling in Secure Messaging: Lifting Security from Sessions to Conversations.** In: *USENIX Security 2023*. Ed. by Joseph A. Calandrino and Carmela Troncoso. USENIX Association, Aug. 2023, pp. 1235–1252.

[DH76] Whitfield Diffie and Martin E. Hellman. **New Directions in Cryptography.** In: *IEEE Transactions on Information Theory* 22.6 (1976), pp. 644–654. DOI: 10.1109/TIT.1976.1055638.

[DJSS18] David Derler, Tibor Jager, Daniel Slamanig, and Christoph Striecks. **Bloom Filter Encryption and Applications to Efficient Forward-Secret 0-RTT Key Exchange.** In: *EUROCRYPT 2018, Part III*. Ed. by Jesper Buus Nielsen and Vincent Rijmen. Vol. 10822. LNCS. Springer, Cham, 2018, pp. 425–455. DOI: 10.1007/978-3-319-78372-7_14.

References iv

[EHM17] Ksenia Ermoshina, Harry Halpin, and Francesca Musiani. **Can johnny build a protocol? co-ordinating developer and user intentions for privacy-enhanced secure messaging protocols.** In: *European Workshop on Usable Security*. 2017.

[GGSW13] Sanjam Garg, Craig Gentry, Amit Sahai, and Brent Waters. **Witness encryption and its applications.** In: *45th ACM STOC*. Ed. by Dan Boneh, Tim Roughgarden, and Joan Feigenbaum. ACM Press, June 2013, pp. 467–476. DOI: 10.1145/2488608.2488667.

[GM15] Matthew D. Green and Ian Miers. **Forward Secure Asynchronous Messaging from Puncturable Encryption.** In: *2015 IEEE Symposium on Security and Privacy*. IEEE Computer Society Press, May 2015, pp. 305–320. DOI: 10.1109/SP.2015.26.

[Gol04] Oded Goldreich. **Foundations of Cryptography: Basic Applications.** Vol. 2. Cambridge, UK: Cambridge University Press, 2004. ISBN: 978-0511721656. DOI: 10.1017/CBO9780511721656.

[Gol97] Oded Goldreich. On the Foundations of Modern Cryptography (Invited Lecture). In: *CRYPTO'97*. Ed. by Burton S. Kaliski Jr. Vol. 1294. LNCS. Springer, Berlin, Heidelberg, Aug. 1997, pp. 46–74. DOI: 10.1007/BFb0052227.

[Gün90] Christoph G. Günther. An Identity-Based Key-Exchange Protocol. In: *EUROCRYPT'89*. Ed. by Jean-Jacques Quisquater and Joos Vandewalle. Vol. 434. LNCS. Springer, Berlin, Heidelberg, Apr. 1990, pp. 29–37. DOI: 10.1007/3-540-46885-4_5.

[Kra05] Hugo Krawczyk. HMQV: A High-Performance Secure Diffie-Hellman Protocol. In: *CRYPTO 2005*. Ed. by Victor Shoup. Vol. 3621. LNCS. Springer, Berlin, Heidelberg, Aug. 2005, pp. 546–566. DOI: 10.1007/11535218_33.

[RSA78] Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman. **A Method for Obtaining Digital Signatures and Public-Key Cryptosystems.** In: *Communications of the Association for Computing Machinery* 21.2 (Feb. 1978), pp. 120–126. DOI: 10.1145/359340.359342.

[RST01] Ronald L. Rivest, Adi Shamir, and Yael Tauman. **How to Leak a Secret.** In: *ASIACRYPT 2001*. Ed. by Colin Boyd. Vol. 2248. LNCS. Springer, Berlin, Heidelberg, Dec. 2001, pp. 552–565. DOI: 10.1007/3-540-45682-1_32.

[Sch90] Claus-Peter Schnorr. **Efficient Identification and Signatures for Smart Cards.** In: *CRYPTO'89*. Ed. by Gilles Brassard. Vol. 435. LNCS. Springer, New York, Aug. 1990, pp. 239–252. DOI: 10.1007/0-387-34805-0_22.