NTNU | Norwegian University of Science and Technology

FROM ENIGMA TO CRYPTO WARS

TTM4205 – Lecture 2

Tjerand Silde

18.08.2026

Contents

A Short History

An Old Cipher

Crypto AG

Crypto Wars

Chat Control

Analyzing Crypto in the Wild

Disclosure and Ethics

Crypto in the Courtroom

Contents

A Short History

An Old Cipher

Crypto AG

Crypto Wars

Chat Control

Analyzing Crypto in the Wild

Disclosure and Ethics

Crypto in the Courtroom

Cryptography Used to Mean One Thing

For most of its history, cryptography meant *secrecy*, and nothing else. Caesar, Vigenère, Enigma, Purple: all of them answer a single question, namely how to stop someone else from reading a message.

The threat model was equally simple. The adversary was a state, the channel was a courier or a radio, and the users were soldiers and diplomats.

Confidentiality Is Not Enough

Confidentiality Is Not Enough

- Encryption hides content, but does not prevent attackers from *changing* it

Confidentiality Is Not Enough

- Encryption hides content, but does not prevent attackers from *changing* it
- A one-time pad is perfectly secret, and an attacker who flips a ciphertext bit flips the corresponding plaintext bit

Confidentiality Is Not Enough

- Encryption hides content, but does not prevent attackers from *changing* it
- A one-time pad is perfectly secret, and an attacker who flips a ciphertext bit flips the corresponding plaintext bit
- Stream ciphers and CTR mode inherit exactly the same problem

Confidentiality Is Not Enough

- Encryption hides content, but does not prevent attackers from *changing* it
- A one-time pad is perfectly secret, and an attacker who flips a ciphertext bit flips the corresponding plaintext bit
- Stream ciphers and CTR mode inherit exactly the same problem
- Padding, error messages, and timing then leak the rest, which is what padding oracles exploit

Confidentiality Is Not Enough

- Encryption hides content, but does not prevent attackers from *changing* it
- A one-time pad is perfectly secret, and an attacker who flips a ciphertext bit flips the corresponding plaintext bit
- Stream ciphers and CTR mode inherit exactly the same problem
- Padding, error messages, and timing then leak the rest, which is what padding oracles exploit

So we need *integrity*: the guarantee that what you decrypt is what was sent.

Integrity and Authenticity

- **Hash functions** give a short fingerprint of a message, but anyone can recompute one, so they authenticate nothing on their own
- **MACs** add a shared secret key, so the receiver knows the message came from someone holding that key, and has not been modified
- **Digital signatures** use public keys instead, so *anyone* can verify, and only the key holder can produce them. That also gives non-repudiation
- **Authenticated encryption** (AES-GCM, ChaCha20-Poly1305) packages confidentiality and integrity into one primitive that is hard to misuse

Most of the failures in this course happen where these are combined badly rather than where a primitive is broken.

Then It Kept Going

Once we could do secrecy, integrity, and authenticity, the field started asking for much stranger properties:

- Key exchange over a public channel, and forward secrecy afterwards
- Zero-knowledge proofs: prove you know a secret, without revealing it
- Commitments, threshold and multi-party computation, secret sharing
- Computing on encrypted data with homomorphic encryption
- Proving that a computation was done correctly, succinctly, to anyone

Where It Ends Up

These are not exotic anymore. They are deployed, at scale:

- **End-to-end encrypted messaging**: Signal, WhatsApp, iMessage, with forward secrecy and post-compromise security for billions of users
- **Blockchains**: signatures, hash chains, and increasingly zero-knowledge proofs for scaling and privacy
- **Electronic voting**: homomorphic tallying and verifiable mix-nets, so that a result can be checked without opening a single ballot
- **Homomorphic encryption and MPC**: computing on medical or financial data that no single party is allowed to see
- **Private measurement**: telemetry and statistics w/o individual records

The Adversary Changed Too

The properties grew, but so did the threat model. The attacker is no longer only someone on the wire:

- The **service provider** itself: end-to-end encryption protects your messages *from the company carrying them*
- The **cloud**, holding your files, backups, and keys
- Whoever **holds your unlocked device**, at a border or after a theft
- The **manufacturer**, through firmware, updates, and standards
- Anyone who can **measure** your device: power, timing, cache
- Client-side scanning puts the provider back into the channel

What This Course Is About

Every one of those systems is built from primitives that are, mathematically, in excellent shape. We are not going to break AES or the dlog problem.

We are going to break the space between the mathematics and the deployment: the implementation, the parameters, the randomness, the composition, and the API.

That is where real systems actually fail, and it is why the history above matters. Each new property added a new way to get it wrong.

Contents

A Short History

An Old Cipher

Crypto AG

Crypto Wars

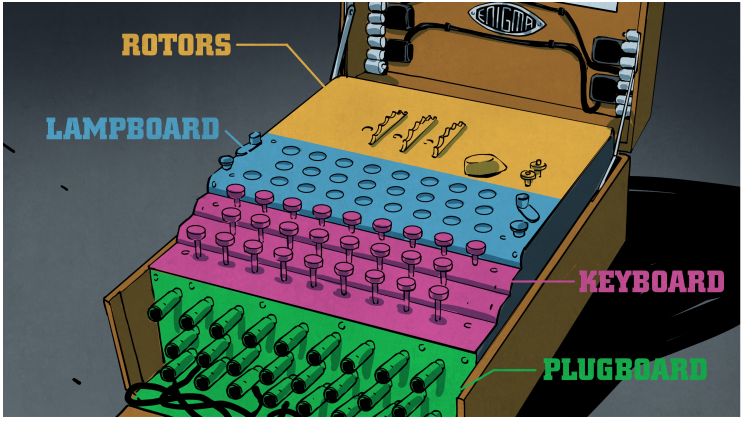
Chat Control

Analyzing Crypto in the Wild

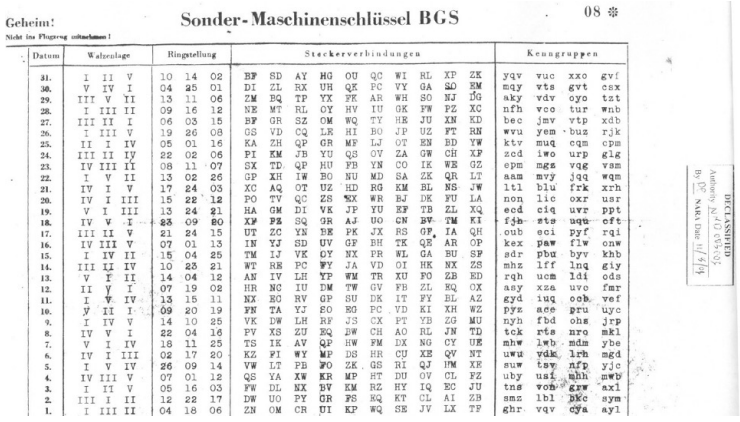
Disclosure and Ethics

Crypto in the Courtroom

Enigma Machine



Code Table



Geheim!

Nicht ins Flugzeug mitnehmen!

Sonder-Maschinenschlüssel BGS

08 ✻

Datum	Walzenlage	Ringstellung	Steckerverbindungen	Kenngruppen
31.	I II V	10 14 02	BF SD AY HG OU QC WI RL XP ZK	yqv vuc xxo gvf
30.	V IV I	04 25 01	DI ZL RX UH QK PC VY GA SO EM	mqy vts gvt csx
29.	III V II	13 11 06	ZM BQ TP YX FE AR WH SO NJ DG	aky vdv oyo tzt
28.	I III II	09 16 12	NE MT RL OY HV IU GK FW PZ XC	nfh vco tur wnb
27.	III II I	06 03 15	BF GR SZ OM WQ TV HE JU XN KD	bec jmv vtp xdb
26.	I III V	19 26 08	GS VD CQ LR HI BO JP UZ FT RN	wvu yem buz rjk
25.	II I IV	05 01 16	KA ZH QP GR MF LJ OT EN BD YW	ktv muq cqm cpm
24.	III II IV	22 02 06	PI KM JB YU QS OV ZA GW CH XF	zcd iwo urp glg
23.	IV III II	08 11 07	SX TD QP HU FB YN CO IK WE GZ	epm mgx vqg vsm
22.	I V II	13 02 26	GP XH IW BO NU MD SA ZK QR LT	aam mvy jqq wqm
21.	IV I V	17 24 03	XC AQ OT UZ HD RG KM BL NS JW	ltl blu frk xrh
20.	IV I III	15 22 12	PO TV QC ZS SX WR BJ DK FU LA	non lic oxr usr
19.	V I III	13 24 21	HA GM DI VK JP YU EF TB ZL XQ	ecd ciq uvr ppt
18.	IV V I	23 09 80	XP PZ SQ GR AJ UO GW BV TM KI	fjb sts uqu cft
17.	III II V	21 24 15	UT ZC YN IE PK JX RS GP IA QH	oub sci pyf rqi
16.	IV III V	07 01 13	IN YJ SD UV GF BH TK QE AR OP	kex paw flw onw
15.	I IV II	15 04 25	TM IJ VK OY NX PR WL GA BU SF	sdr pbu byv khb
14.	III II IV	10 23 21	WT RE PC WY JA VD OI HK NX ZS	mhz lff lnq giy
13.	V I II	14 04 12	AN IV LH YP WM TR XU FO ZB ED	rqh ucm ldi ods
12.	II V I	07 19 02	HR NG IU DM TW GV FB ZL EQ OX	asy xza uvo fmr
11.	I V IV	13 15 11	NX EC RV GP SU DK IT FY BL AZ	gyd iuq oob vef
10.	V II I	09 20 19	FN TA YJ SO EG PC VD KI XH WZ	pyz ace pru uyc
9.	I IV V	14 10 25	VK DW LH RF JS CX PT YB ZG MU	nyh fbd ohs jrp
8.	IV V I	22 04 16	PV XS ZU EQ DW CH AO RL JW TD	tck rts nro mkl
7.	V I IV	18 11 25	TS IK AV QP HW FM DX NG CY UE	mhw lwb mdm ybe
6.	IV I III	02 17 20	KZ FI WY MP DS HR CU XE QV NT	uwu vdk lrh mgd
5.	I V IV	26 09 14	VW LT PB FO ZK GS RI QJ HM XE	suw tsv nfp yjc
4.	IV III V	07 01 12	QS YA XW KR MP HT DU OV CL FZ	uby usk mhh mwb
3.	I II V	05 16 03	FW DL NX BV KM BZ HY IQ EC JU	tns vob grw axl
2.	III I II	12 22 17	DW UO PY GR FS SQ KT CL AI ZB	smz lbl bKc sym
1.	I III II	04 18 06	ZN OM CR UI KP WQ SE JV LX TF	ghr vqv cya ayl

Security of Enigma

Security of Enigma

- Choose three rotors out of five

Security of Enigma

- Choose three rotors out of five
- Each rotor has 26 starting positions

Security of Enigma

- Choose three rotors out of five
- Each rotor has 26 starting positions
- Plugboard connecting ten letter-pairs

Security of Enigma

- Choose three rotors out of five
- Each rotor has 26 starting positions
- Plugboard connecting ten letter-pairs
- Leads to roughly 2^{67} possible settings, of which the plugboard alone contributes about 2^{47}

Security of Enigma

- Choose three rotors out of five
- Each rotor has 26 starting positions
- Plugboard connecting ten letter-pairs
- Leads to roughly 2^{67} possible settings, of which the plugboard alone contributes about 2^{47}
- Its operators believed it to be unbreakable

Flaws of Enigma

Flaws of Enigma

- A plaintext letter can never be encrypted to itself

Flaws of Enigma

- A plaintext letter can never be encrypted to itself
- They had access to known plaintexts every day

Flaws of Enigma

- A plaintext letter can never be encrypted to itself
- They had access to known plaintexts every day
- Each contradiction removed millions of settings

Flaws of Enigma

- A plaintext letter can never be encrypted to itself
- They had access to known plaintexts every day
- Each contradiction removed millions of settings
- It took two hours to brute force a key each day

Flaws of Enigma

- A plaintext letter can never be encrypted to itself
- They had access to known plaintexts every day
- Each contradiction removed millions of settings
- It took two hours to brute force a key each day
- Rejewski and the Polish Cipher Bureau broke it first in 1932, and handed their work to Britain in 1939

Flaws of Enigma

- A plaintext letter can never be encrypted to itself
- They had access to known plaintexts every day
- Each contradiction removed millions of settings
- It took two hours to brute force a key each day
- Rejewski and the Polish Cipher Bureau broke it first in 1932, and handed their work to Britain in 1939
- Turing and the team at Bletchley Park then broke the wartime versions in 1941/42

Facts about Enigma

Facts about Enigma

- The UK disclosed that they broke it in 1970

Facts about Enigma

- The UK disclosed that they broke it in 1970
- There are roughly 300 (publicly known) copies

Facts about Enigma

- The UK disclosed that they broke it in 1970
- There are roughly 300 (publicly known) copies
- Some versions of Enigma have four rotors

Facts about Enigma

- The UK disclosed that they broke it in 1970
- There are roughly 300 (publicly known) copies
- Some versions of Enigma have four rotors
- The auction value is between 3 and 5 MNOK

Enigma at NTNU

Contents

A Short History

An Old Cipher

Crypto AG

Crypto Wars

Chat Control

Analyzing Crypto in the Wild

Disclosure and Ethics

Crypto in the Courtroom

Crypto AG

A Swiss company named Crypto AG, funded by the Swede named Boris Hagelin, sold encryption machines to nation states all over the world after the second world war. They were similar to the Enigma machine.

Widespread Usage

THE AMERICAS
Argentina
Brazil
Chile
Colombia
Honduras
Mexico
Nicaragua
Peru
Uruguay
Venezuela

EUROPE
Austria
Czechoslovakia
Greece
Hungary
Ireland
Italy
Portugal
Romania
Spain
Turkey
Vatican City
Yugoslavia

AFRICA
Algeria
Angola
Egypt
Gabon
Ghana
Guinea
Ivory Coast
Libya
Mauritius
Morocco
Nigeria
Rep. of the Congo
South Africa
Sudan
Tanzania
Tunisia
Zaire
Zimbabwe

MIDDLE EAST
Iran
Iraq
Jordan
Kuwait
Lebanon
Oman
Qatar
Saudi Arabia
Syria
U.A.E.

REST OF ASIA
Bangladesh
Burma
India
Indonesia
Japan
Malaysia
Pakistan
Philippines
South Korea
Thailand
Vietnam

WORLDWIDE ORGANIZATION
United Nations

The records show that at least four countries — Israel, Sweden, Switzerland and the United Kingdom — were aware of the operation or were provided intelligence from it by the United States or West Germany.

CIA Backdoor

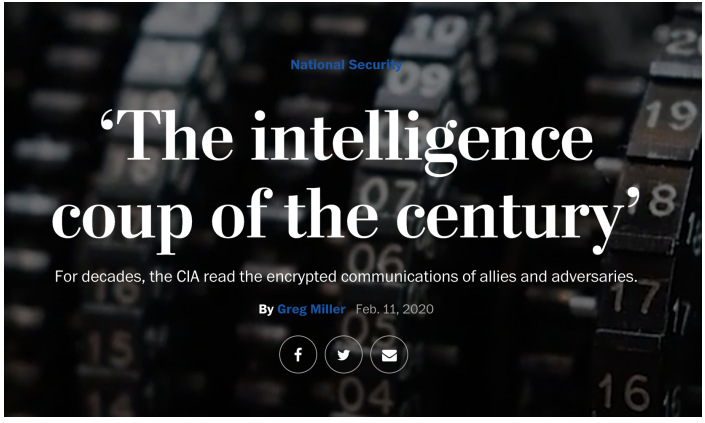


Figure: https://www.washingtonpost.com/graphics/2020/world/national-security/cia-crypto-encryption-machines-espionage

Contents

A Short History

An Old Cipher

Crypto AG

Crypto Wars

Chat Control

Analyzing Crypto in the Wild

Disclosure and Ethics

Crypto in the Courtroom

Crypto Wars

Essentially 30+ year ongoing debate between policymakers and technologists about encryption and surveillance

Typically portrayed as "Safety" vs. "Privacy" to get "Security"

Crypto War I

Crypto War I

- The 1990s: Wire-tapping vs. cryptography

Crypto War I

- The 1990s: Wire-tapping vs. cryptography
- AT&T phone calls encrypted with DH and DES

Crypto War I

- The 1990s: Wire-tapping vs. cryptography
- AT&T phone calls encrypted with DH and DES
- The Clipper-chip and key escrow (broken)

Crypto War I

- The 1990s: Wire-tapping vs. cryptography
- AT&T phone calls encrypted with DH and DES
- The Clipper-chip and key escrow (broken)
- Law vs. technology, export control, free speech

Crypto War I

- The 1990s: Wire-tapping vs. cryptography
- AT&T phone calls encrypted with DH and DES
- The Clipper-chip and key escrow (broken)
- Law vs. technology, export control, free speech
- EFF DES cracker broke 56 bit DES in 1998

Crypto War I

- The 1990s: Wire-tapping vs. cryptography
- AT&T phone calls encrypted with DH and DES
- The Clipper-chip and key escrow (broken)
- Law vs. technology, export control, free speech
- EFF DES cracker broke 56 bit DES in 1998
- The US government allows crypto from $\sim$ 2000

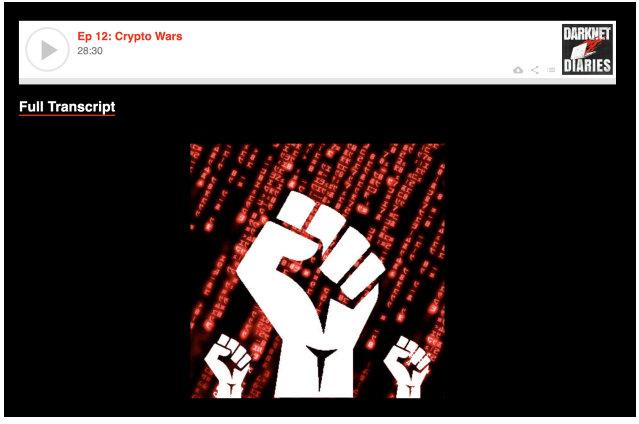


Figure: https://darknetdiaries.com/episode/12

Crypto War II

Crypto War II

- Roughly the years 2010-2016: The "Going dark" debate

Crypto War II

- Roughly the years 2010-2016: The "Going dark" debate
- Communication (calls, messages) usually not encrypted

Crypto War II

- Roughly the years 2010-2016: The "Going dark" debate
- Communication (calls, messages) usually not encrypted
- The data stored on phones and laptops was encrypted

Crypto War II

- Roughly the years 2010-2016: The "Going dark" debate
- Communication (calls, messages) usually not encrypted
- The data stored on phones and laptops was encrypted
- The 2013 Snowden revelations and mass surveillance

Crypto War II

- Roughly the years 2010-2016: The "Going dark" debate
- Communication (calls, messages) usually not encrypted
- The data stored on phones and laptops was encrypted
- The 2013 Snowden revelations and mass surveillance
- Crypto vs Mass Surveillance `http://cms16.item.ntnu.no`

Crypto War II

- Roughly the years 2010-2016: The "Going dark" debate
- Communication (calls, messages) usually not encrypted
- The data stored on phones and laptops was encrypted
- The 2013 Snowden revelations and mass surveillance
- Crypto vs Mass Surveillance `http://cms16.item.ntnu.no`
- The FBI vs. Apple case and breaking into devices

Crypto War II

- Roughly the years 2010-2016: The "Going dark" debate
- Communication (calls, messages) usually not encrypted
- The data stored on phones and laptops was encrypted
- The 2013 Snowden revelations and mass surveillance
- Crypto vs Mass Surveillance `http://cms16.item.ntnu.no`
- The FBI vs. Apple case and breaking into devices
- Standardized crypto backdoored by NSA (Lecture 6)

Figure: `https://youtu.be/bB68G8tLh38`

The Moral Character of Cryptographic Work*

Phillip Rogaway

Department of Computer Science
University of California, Davis, USA
rogaway@cs.ucdavis.edu

December 2015
(minor revisions March 2016)

Figure: https://web.cs.ucdavis.edu/~rogaway/papers/moral-fn.pdf

Crypto War III

Crypto War III

- 2017-now: The ongoing "Safety vs. Privacy" debate

Crypto War III

- 2017-now: The ongoing "Safety vs. Privacy" debate
- EARN IT ACT, ONLINE SAFETY BILL, CHAT CONTROL 2.0

Crypto War III

- 2017-now: The ongoing "Safety vs. Privacy" debate
- EARN IT ACT, ONLINE SAFETY BILL, CHAT CONTROL 2.0
- Age verification, content scanning, liable providers

Crypto War III

- 2017-now: The ongoing "Safety vs. Privacy" debate
- EARN IT ACT, ONLINE SAFETY BILL, CHAT CONTROL 2.0
- Age verification, content scanning, liable providers
- The battleground moved from the key to the endpoint

Crypto War III

- 2017-now: The ongoing "Safety vs. Privacy" debate
- EARN IT ACT, ONLINE SAFETY BILL, CHAT CONTROL 2.0
- Age verification, content scanning, liable providers
- The battleground moved from the key to the endpoint

The next section looks at where that fight stands today.

Keys Under Doormats:

MANDATING INSECURITY BY REQUIRING GOVERNMENT ACCESS TO ALL DATA AND COMMUNICATIONS

Harold Abelson, Ross Anderson, Steven M. Bellovin, Josh Benaloh, Matt Blaze, Whitfield Diffie, John Gilmore, Matthew Green, Susan Landau, Peter G. Neumann, Ronald L. Rivest, Jeffrey I. Schiller, Bruce Schneier, Michael Specter, Daniel J. Weitzner

Figure:
https://www.schneier.com/wp-content/uploads/2016/09/paper-keys-under-doormats-CSAIL.pdf

Contents

A Short History

An Old Cipher

Crypto AG

Crypto Wars

Chat Control

Analyzing Crypto in the Wild

Disclosure and Ethics

Crypto in the Courtroom

Chat Control

The European Commission's 2022 proposal on child sexual abuse material would oblige online services to detect, report, and remove prohibited content, in practice through *client-side scanning*: checking messages on your own device, before they are encrypted.

It is the current front of the Crypto Wars, and it is being decided while you take this course.

Why Cryptographers Object

- Scanning content before encryption does not preserve end-to-end encryption, it works around it: E2EE becomes a conditional privilege rather than a guarantee
- There is no backdoor that only the good guys can use. Scanning code on every device is a new, high-value target: disable it, subvert it, or plant evidence with it
- The database of what to detect is itself a risk. If it leaks, criminals learn what is detected; if entries can be injected, anyone's benign content can be flagged
- Detection orders applied to everyone amount to general monitoring, which sits badly with the European Court of Human Rights ruling in *Podchasov v. Russia*

The Numbers

No classifier is perfect, and at this scale imperfect is expensive. A European Parliament research study concluded that no existing technology detects such material without error rates that would wrongly flag large amounts of lawful communication.

Irish police figures illustrate it: of 4192 reports from service providers in one period, about 852 (20%) were actual abuse material, and 471 (11%) were confirmed false positives.

Scaled to European messaging traffic, that is millions of false positives, each one a real person whose private messages are read by strangers.

Who Is Exempt

Recent Council drafts exclude non-public communication used for national security, government, and professional secrecy.

The justification is the confidentiality of classified information. The effect is that the people writing the rules keep their own confidential channels, while everyone else is scanned.

If a measure is safe enough for the public, it should be safe enough for its authors. That asymmetry is itself an argument about how well the risks are understood.

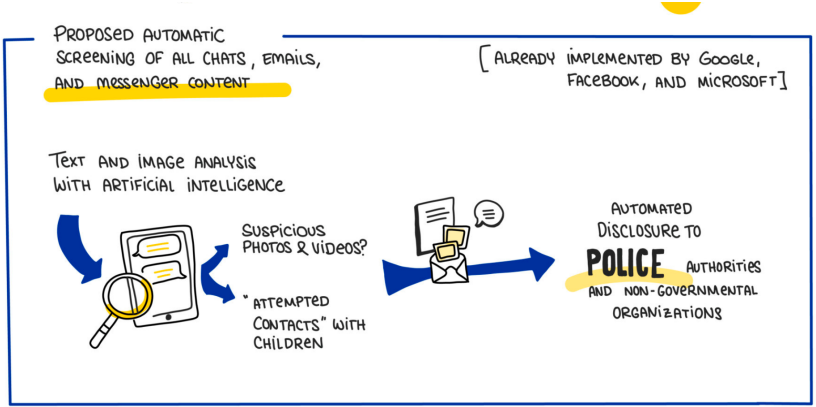


Figure: https://www.patrick-breyer.de/en/posts/chat-control

EU vil overvåke chat-meldingene dine

Ville du godtatt at staten leser meldinger du sender og ser gjennom bildene du har på mobilen? Snart kan det hende du ikke har noe valg.

Disse meldings- og skylagrings-appene er blant de som kan bli rammet av lovforslaget som EU prøver å få godkjent.

FOTO: GEIR EVENSEN / NRK

Geir Evensen
Journalist

Publisert 14. aug. kl. 08:52
Oppdatert 14. aug. kl. 09:22

Figure: https://www.nrk.no/urix/chat-control_-eu-vil-masseovervake-innbyggerne-1.17983785

What Would Actually Help

Aranha and Melissaris point to alternatives that do not require general monitoring:

- Targeted, warrant-based operations against suspects' devices
- Disruption on the perpetrator side, where the material is produced and shared
- Opt-in safety features, especially for accounts belonging to minors
- Properly funding the investigators who already cannot process the reports they get today

You will be asked about this in your career, by employers, journalists, and politicians. It is worth having thought it through.

Contents

A Short History

An Old Cipher

Crypto AG

Crypto Wars

Chat Control

Analyzing Crypto in the Wild

Disclosure and Ethics

Crypto in the Courtroom

Cryptography in the Wild

Most of this course is about cryptography as it is actually deployed, not as it is specified. Albrecht and Paterson call this *analysing cryptography in the wild*: take a real protocol, implementation, or product, work out what it really does, and then try to break it or prove it secure.

They have done this to SSH, TLS, Telegram, MEGA, and more, and have written up how they go about it.

Albrecht and Paterson, *Analysing Cryptography in the Wild: A Retrospective*, https://eprint.iacr.org/2024/532

How to Choose an Object of Study

Two ways in:

- **From the crypto:** you know an attack technique and look for deployments that might be vulnerable to it
- **From the target:** you care about a specific system, often because many people depend on it, and you go looking for whatever is there

Telegram came from the second kind: widely used by people at risk, a bespoke protocol (MTProto 2.0) instead of TLS, and little prior analysis.

Note that surviving a security audit says little, as audits are usually shallow, time-boxed, and scoped by the vendor.

The Method

- **Pick an adversarial model.** Who is attacking, and what can they do? This choice is where most of the difficulty lives, and getting it wrong invalidates the result
- **Ingest.** Read specifications, source code, and clients. Different clients often do different checks, and the documentation rarely matches the code
- **Model.** Write pseudocode for what the system actually does, at a level of abstraction that keeps the security-relevant details
- **Attack or prove.** Look for the classics first, then chain them or invent new ones. Or prove security in your model, and be explicit about the assumptions

What to Look For First

The usual suspects, straight from the retrospective:

- ECB or home-made modes
- No integrity at all
- Mac-then-Encrypt, Encrypt-and-Mac
- Padding oracles
- Nonce reuse
- Key reuse across purposes
- No domain separation
- Bad interaction between protocols
- Weak or home-brew PRNGs
- Compression with encryption

Every one of these shows up somewhere in this course.

Abstractions Are Where You Get Burned

A proof is not a binary statement about a system. It is a statement about a *model* of that system, in a particular adversarial model, under particular assumptions.

Albrecht and Paterson are candid about this: their own SSH proof missed the Terrapin attack, because a simplifying assumption in the model turned out to be both false and significant.

So when someone says “it is proven secure”, the useful questions are: proven about which model, against which adversary, and which parts of the real system were abstracted away?

Proofs of Concept

It is far easier to convince a vendor, a journalist, or a program committee with a working exploit than with an argument, and most bug bounty programmes require one.

But building one is laborious, and the work is usually not the interesting cryptography: it is wrestling with the state machine, the parser, and the encoding. Sometimes it is a 2^{60} computation and several person-months.

A Worked Example: MEGA

MEGA is a cloud storage platform with more than 265 million accounts, advertising end-to-end encryption: your files should stay confidential *even if MEGA itself is malicious*. That is a strong threat model, and a high-value target.

Backendal, Haller, and Paterson broke it in 2022. MEGA patched. Albrecht, Haller, Mareková, and Paterson then broke the patch.

Albrecht, Haller, Mareˇková, and Paterson, *Caveat Implementor! Key Recovery Attacks on MEGA*, https://eprint.iacr.org/2023/329

Every Item on the Checklist

Look at what actually went wrong, against the list from earlier:

- **No key separation**: one master key encrypts both the RSA private key and the file keys
- **No integrity on stored keys**: a MAC would have stopped every one of these attacks
- **ECB mode**, giving an encryption oracle in an unrelated feature
- **Verbose errors**, turning implementation details into a remote side channel
- **Buggy big-integer arithmetic**, where errors became oracles

The cryptography is standard. The composition is what fails.

Patches Are Attack Surface Too

MEGA responded to the first attacks with lightweight sanity checks on the RSA private key. Reasonable-looking, and cheap.

The follow-up work attacks *the sanity checks themselves*: the different ways they fail become distinguishable, and the resulting oracles recover the private key at only slightly higher cost than before. One attack exploits how the code handles $u = q^{-1} \bmod p$; the other is a small subgroup attack.

A patch without a security model is a new implementation, with new bugs.

Attacks Only Get Better

- The original key recovery needed an estimated 512 logins, which MEGA used to argue it was impractical
- Ryan and Heninger reduced it to 6
- This paper reduced it to 2

“Not practical” is a statement about today’s attack, not about the system. If your defence rests on an attack’s cost, expect that number to fall.

The Target Moved Up the Stack

They make an important point: it can look as though the golden age of attacks on deployed cryptography is over, given the rigour that went into TLS 1.3.

What actually happened is that the target moved. As applications go beyond protecting data in transit or at rest, more complex cryptographic systems get deployed at scale, often with little input from the cryptographic community.

That is the gap this course lives in, and it is where your essay topics come from.

Contents

A Short History

An Old Cipher

Crypto AG

Crypto Wars

Chat Control

Analyzing Crypto in the Wild

Disclosure and Ethics

Crypto in the Courtroom

Coordinated Vulnerability Disclosure

You found a break. Now what?

- Tell the vendor privately, with a deadline, commonly 90 days
- If nothing is fixed by then, publish anyway
- Before the deadline, coordinate the public disclosure with the vendor
- Keep the decision on timing under *your* control

The 90 days is arbitrary. It works for software with an established update process, and is unrealistic for hardware and microarchitectural flaws.

Responsible to Whom?

A common misconception is that you owe the vendor something. Usually there is no legal or commercial relationship at all.

Albrecht and Paterson argue that the responsibility is to the *people who have to rely on the technology*. Sometimes those users are better served by being warned off a protocol than by a rushed patch from an inexperienced team, but often they have no realistic alternative, which is why cooperating with the vendor is usually still the right strategy.

Decide who you consider yourself responsible to *before* you disclose.

How It Is Received

- **Vendors.** Often have no disclosure process at all. Some patch loudly in public repositories, which lets others reverse-engineer the bug before users are protected
- **Academia.** Rewards novelty, which sits awkwardly with work whose value is that it protects real users from a mundane bug
- **The public.** A short burst of reporting, and then some variant of “don’t roll your own crypto”

That slogan is true but unhelpful: it hides what actually lets someone roll their own, namely careful modelling, formal analysis, and secure implementation.

Be Honest About Impact

You also have a responsibility not to oversell. Insinuating that a robust system is broken can push people onto genuinely insecure alternatives.

Say precisely what is broken, under which assumptions, and what an attacker actually gains. “Key recovery given one chosen ciphertext and physical access” and “we can read your messages” are very different claims.

This is how we grade: the claim you can defend, not the one that sounds best.

Cryptography Is Political

Rogaway's argument, which we saw earlier: cryptographic work is not neutral. What you choose to work on, whose threat model you adopt, and what you leave unstudied are all political choices, whether or not you make them consciously.

The Crypto Wars are the obvious case, but the same holds for a scheme that quietly assumes an honest server, or a protocol whose threat model excludes the people most at risk.

Ethics in Our Own Work

- Who is harmed if you are right, and who is harmed if you stay quiet?
- Do you have permission to test the system you are testing? Attacking live services or other people's devices is a crime, however good the intent
- Are you studying people, and not just protocols? Then you need consent, ethical approval, and care with the data
- Are you honest about limitations, including the ones that make your result less exciting?
- Whose problem did you choose to work on, and who could have used the help more?

For this course: attack only the equipment and the targets that we hand you.

Undone Science

Aranha and Melissaris borrow a useful term: *undone science*, the research that is systematically not done because nobody is funded, incentivised, or politically served by doing it.

Their claim is that the questions we raised earlier about Chat Control, whether large-scale content scanning can work at all and at what cost to rights, are exactly such undone science, deferred in the name of technological urgency.

Noticing what is missing from a debate is part of the job, and choosing to do the missing work is one of the most useful things a researcher can do.

Aranha and Melissaris, *What is Cryptography Hiding from Itself?*,
https://eprint.iacr.org/2025/1951

Contents

A Short History

An Old Cipher

Crypto AG

Crypto Wars

Chat Control

Analyzing Crypto in the Wild

Disclosure and Ethics

Crypto in the Courtroom

Cryptography Ends Up in Court

Break a system in a paper and the consequence is a patch. Break one as a police force and the consequence is a prosecution, where the evidence, the method, and eventually the cryptography are argued over in a courtroom.

The clearest case we have is **Encrochat**; documented in detail only this year.

Albrecht, Park, Specter, and Stebila, *A Real-World Law-Enforcement Hack: The Case of Encrochat*, CRYPTO 2026, https://eprint.iacr.org/2026/1319

What Encrochat Was

- A vertically integrated service: the same company sold the phone *and* ran the service
- Modified Android handsets, sold only through resellers, with the camera, microphone, GPS, and USB data removed
- Its own PKI, full disk encryption, remote and local wipe
- Encrypted messaging built on the **Signal protocol**, so the cryptography itself was reasonable
- Around 66 000 users by 2020, heavily used by organised crime

Note what this means: the protocol was not the weak point.

How It Was Compromised

- French investigators obtained court orders compelling the hosting provider and the domain registrar to cooperate
- They compromised the servers, then pushed **malware to the handsets** as if it were a normal update
- The malware exfiltrated historical and live messages and metadata over months, in 2020
- The end-to-end encryption was never broken. It was *side-stepped*, at the endpoint
- The data supported over 6 000 arrests and prosecutions across Europe

This is what a state adversary actually does, and it is not cryptanalysis.

What the Courts Had to Decide

- In France, the technical operation was run by the intelligence service, so its details became a **national defence secret**, kept even from the judge who authorised it
- Defendants argued they could not mount a defence against evidence whose provenance they were not allowed to examine. The Conseil Constitutionnel ruled the secrecy constitutional
- The Cour de Cassation nevertheless found that the required certificate of authenticity was missing in at least one prosecution
- In the UK, admissibility turned on whether the malware performed *interception* or *equipment interference*, that is, whether data was taken in transit or at rest

Was Everyone a Criminal?

The warrants argued that “Encrochat devices are used exclusively by criminals”, and that claim justified compromising the entire network rather than targeting suspects individually.

The UK operation examined about 300 devices, roughly 4% of users, that contained no evidence of criminality. The assessment did not change.

The same argument would be trivially false for WhatsApp, Signal, or Tor. Deciding when a whole network may be compromised is a question about proportionality, not about cryptography.

Open Problems for Us

The authors point at two gaps in how cryptographers model this:

- **Subversion.** Our literature models algorithm substitution attacks, but Encrochat was bypassed entirely, through the update channel. Securing binaries against third-party tampering, with reproducible builds and binary transparency, is the open problem
- **Covert adversaries.** Our adversary is an algorithm. This one was an organization: it depended on third parties, the registrar made a mistake that nearly exposed the operation, and an analyst tipped off a contact and was convicted for it

An adversary made of people fails in ways no security definition captures.

Questions?