User empowerment for security and privacy in IoT

Chloe Martindale

Technische Universiteit Eindhoven

joint with CEA, University of Murcia, and IBM Research Zurich

Primary use case: smart cars

ETSI [1] compiled a list of mandatory use cases of smart cars, for example:

Primary use case: smart cars

ETSI [1] compiled a list of mandatory use cases of smart cars, for example:

- A stationary vehicle in a dangerous location sends out warnings to other vehicles

Primary use case: smart cars

ETSI [1] compiled a list of mandatory use cases of smart cars, for example:

- A stationary vehicle in a dangerous location sends out warnings to other vehicles
- Vehicles that detect hazardous conditions (due to weather, traffic, etc.) send out warnings to other vehicles

Primary use case: smart cars

ETSI [1] compiled a list of mandatory use cases of smart cars, for example:

- A stationary vehicle in a dangerous location sends out warnings to other vehicles
- Vehicles that detect hazardous conditions (due to weather, traffic, etc.) send out warnings to other vehicles
- Emergency vehicles constantly broadcast their position

Primary use case: smart cars

ETSI [1] compiled a list of mandatory use cases of smart cars, for example:

- A stationary vehicle in a dangerous location sends out warnings to other vehicles
- Vehicles that detect hazardous conditions (due to weather, traffic, etc.) send out warnings to other vehicles
- Emergency vehicles constantly broadcast their position

This increases the safety of our roads,

Primary use case: smart cars

ETSI [1] compiled a list of mandatory use cases of smart cars, for example:

- A stationary vehicle in a dangerous location sends out warnings to other vehicles
- Vehicles that detect hazardous conditions (due to weather, traffic, etc.) send out warnings to other vehicles
- Emergency vehicles constantly broadcast their position

This increases the safety of our roads, but we frequently broadcast our position to anyone who wants to know!

Primary use case: smart cars

ETSI [1] compiled a list of mandatory use cases of smart cars, for example:

- A stationary vehicle in a dangerous location sends out warnings to other vehicles
- Vehicles that detect hazardous conditions (due to weather, traffic, etc.) send out warnings to other vehicles
- Emergency vehicles constantly broadcast their position

This increases the safety of our roads, but we frequently broadcast our position to anyone who wants to know! Privacy nightmare!

Primary use case: smart cars

Examples of responses:

Primary use case: smart cars

Examples of responses:

- notifying the driver

Primary use case: smart cars

Examples of responses:

- notifying the driver
- sending messages to other parties

Primary use case: smart cars

Examples of responses:

- notifying the driver
- sending messages to other parties
- activating vehicular components (e.g. indicators, brake lights)

Primary use case: smart cars

Examples of responses:

- notifying the driver
- sending messages to other parties
- activating vehicular components (e.g. indicators, brake lights)
- reconfiguring vehicle to reduce crash impact (e.g. deploy airbags)

Primary use case: smart cars

Examples of responses:

- notifying the driver
- sending messages to other parties
- activating vehicular components (e.g. indicators, brake lights)
- reconfiguring vehicle to reduce crash impact (e.g. deploy airbags)
- intervene directly in the driving by applying brakes, steering, or accelerating

Primary use case: smart cars

Examples of responses:

- notifying the driver
- sending messages to other parties
- activating vehicular components (e.g. indicators, brake lights)
- reconfiguring vehicle to reduce crash impact (e.g. deploy airbags)
- intervene directly in the driving by applying brakes, steering, or accelerating

If a malicious third party can send rogue information to the chip..
Security nightmare!

Solution: crypto

How difficult it is to add crypto to such a chip?

Solution: crypto

How difficult it is to add crypto to such a chip? Any crypto needs to be

Solution: crypto

How difficult it is to add crypto to such a chip? Any crypto needs to be

- Fast.

Solution: crypto

How difficult it is to add crypto to such a chip? Any crypto needs to be

- Fast. The ITS requirements are
  - 1000-1600 incoming packets per second
  - 15 outgoing packets per second

Solution: crypto

How difficult it is to add crypto to such a chip? Any crypto needs to be

- Fast. The ITS requirements are
  - 1000-1600 incoming packets per second
  - 15 outgoing packets per second
- The packet delay is limited to 50ms (including preparing outgoing packet, transmission, processing, and verification).

Solution: crypto

How difficult it is to add crypto to such a chip? Any crypto needs to be

- Fast. The ITS requirements are
  - 1000-1600 incoming packets per second
  - 15 outgoing packets per second
- The packet delay is limited to 50ms (including preparing outgoing packet, transmission, processing, and verification).
- Low memory: the data should fit on a small chip and not increase the cost by too much.

Solution: crypto

How difficult it is to add crypto to such a chip? Any crypto needs to be

- Fast. The ITS requirements are
  - 1000-1600 incoming packets per second
  - 15 outgoing packets per second
- The packet delay is limited to 50ms (including preparing outgoing packet, transmission, processing, and verification).
- Low memory: the data should fit on a small chip and not increase the cost by too much.
- Test of time: cars last a long time compared to cryptosystems. So we need to be over cautious!

Interlude: what is pairing-based cryptography?

- Suppose that $(G_1, +)$, $(G_2, +)$, and $(G, \cdot)$ are groups where the *Discrete Logarithm Problem* is hard.

Interlude: what is pairing-based cryptography?

- Suppose that $(G_1, +)$, $(G_2, +)$, and $(G, \cdot)$ are groups where the *Discrete Logarithm Problem* is hard. Examples:
  - For $(G_1, +)$ or $(G_2, +)$, the group of rational points on an elliptic curve over a finite field of > 128 bits.

Interlude: what is pairing-based cryptography?

- Suppose that $(G_1, +)$, $(G_2, +)$, and $(G, \cdot)$ are groups where the *Discrete Logarithm Problem* is hard. Examples:
  - For $(G_1, +)$ or $(G_2, +)$, the group of rational points on an elliptic curve over a finite field of > 128 bits. (This is the basis of *elliptic curve cryptography*).
  - For $(G, \cdot)$, the units of a large finite field $\mathbb{F}_p^*$.

Interlude: what is pairing-based cryptography?

- Suppose that $(G_1, +)$, $(G_2, +)$, and $(G, \cdot)$ are groups where the *Discrete Logarithm Problem* is hard. Examples:
  - For $(G_1, +)$ or $(G_2, +)$, the group of rational points on an elliptic curve over a finite field of > 128 bits. (This is the basis of *elliptic curve cryptography*).
  - For $(G, \cdot)$, the units of a large finite field $\mathbb{F}_p^*$. (This is the basis of traditional RSA).
- A *pairing* is a nice (non-degenerate, bilinear) map

$$G_1 \times G_2 \longrightarrow G.$$

Interlude: what is pairing-based cryptography?

- Suppose that $(G_1, +)$, $(G_2, +)$, and $(G, \cdot)$ are groups where the *Discrete Logarithm Problem* is hard. Examples:
  - For $(G_1, +)$ or $(G_2, +)$, the group of rational points on an elliptic curve over a finite field of > 128 bits. (This is the basis of *elliptic curve cryptography*).
  - For $(G, \cdot)$, the units of a large finite field $\mathbb{F}_p^*$. (This is the basis of traditional RSA).
- A *pairing* is a nice (non-degenerate, bilinear) map

$$G_1 \times G_2 \longrightarrow G.$$

- Via the pairing, we can 'translate' the discrete logarithm problem on G_1 to G.

Why use pairing-based cryptography?

- Using pairings allows for more complicated protocols - making privacy possible!
- Our current project is to improve the state-of-the-art on pairing-based crypto, or develop new tools, to meet the IoT requirements.

Why use pairing-based cryptography?

- Using pairings allows for more complicated protocols - making privacy possible!
- Our current project is to improve the state-of-the-art on pairing-based crypto, or develop new tools, to meet the IoT requirements.
- Each verification takes several pairing computations and several operations in large finite fields.

Why use pairing-based cryptography?

- Using pairings allows for more complicated protocols - making privacy possible!
- Our current project is to improve the state-of-the-art on pairing-based crypto, or develop new tools, to meet the IoT requirements.
- Each verification takes several pairing computations and several operations in large finite fields.
- The fastest pairing works in finite fields of the form $\mathbb{F}_{p^{12}}$ known takes 11ms with current methods.

Why use pairing-based cryptography?

- Using pairings allows for more complicated protocols - making privacy possible!
- Our current project is to improve the state-of-the-art on pairing-based crypto, or develop new tools, to meet the IoT requirements.
- Each verification takes several pairing computations and several operations in large finite fields.
- The fastest pairing works in finite fields of the form $\mathbb{F}_{p^{12}}$ known takes 11ms with current methods.
- Bad news:

Why use pairing-based cryptography?

- Using pairings allows for more complicated protocols - making privacy possible!
- Our current project is to improve the state-of-the-art on pairing-based crypto, or develop new tools, to meet the IoT requirements.
- Each verification takes several pairing computations and several operations in large finite fields.
- The fastest pairing works in finite fields of the form $\mathbb{F}_{p^{12}}$ known takes 11ms with current methods.
- Bad news: this pairing (and all other known fast pairings) may no longer be secure..

Updating pairing-based cryptography

- Due to work of Guillevic, Morain, and Thomé ([4]), pairing based cryptography over $\mathbb{F}_{p^{12}}$ may no longer be secure.

Updating pairing-based cryptography

- Due to work of Guillevic, Morain, and Thomé ([4]), pairing based cryptography over $\mathbb{F}_{p^{12}}$ may no longer be secure.
- A future goal of this project is to see how far their results can be applied.

Updating pairing-based cryptography

- Due to work of Guillevic, Morain, and Thomé ([4]), pairing based cryptography over $\mathbb{F}_{p^{12}}$ may no longer be secure.
- A future goal of this project is to see how far their results can be applied.
- We have developed pairings over $\mathbb{F}_{p^{15}}$ as a compromise (slightly slower, but still secure) ([2]).

Updating pairing-based cryptography

- Due to work of Guillevic, Morain, and Thomé ([4]), pairing based cryptography over $\mathbb{F}_{p^{12}}$ may no longer be secure.
- A future goal of this project is to see how far their results can be applied.
- We have developed pairings over $\mathbb{F}_{p^{15}}$ as a compromise (slightly slower, but still secure) ([2]).
- We have developed a cryptosystem which requires less pairing computations ([3]).

Bibliography

- ETSI, *Intelligent transport systems (its); security; threat, vulnerability and risk analysis (tvra)*, TR 102893, 2010.
- Chuengsatiansup and Martindale, *Pairing-Friendly Twisted Hessian Curves* (in preparation), 2017.
- Camenisch, Drijvers, and Dubovitskaya, *Practical UC-Secure Delegatable Credentials with Attributes and Their Application to Blockchain*, AMS CCS, 2017.
- Guillevic, Morain, and Thomé, *Solving discrete logarithms on a 170-bit MNT curve by pairing reduction*, SAC, 2016.