Lab 2: Smart Thermostat Specification

The Wild and Wonderful World of Thermostats

In the past decade, home automation and "smart home" devices have made their way into our lives. From robot vacuum cleaners to voice-enabled assistants which control things like lighting, speakers and home security systems, they promise to make our everyday lives easier, more comfortable, and even more sustainable. Spare a thought for the humble thermostat, the device which allows you to set the temperature in a room or across several rooms. *Smart* thermostats promise to save you energy (and money!) by heating and cooling the right parts of your home only when you actually need them to be heated and by figuring out the most energy efficient way to do so. At least that's the pitch.

To give you some context, here's a genealogy (if you will) of the kinds of thermostats you may find in homes today, and how they work.

- **Basic thermostats** allow users to set the temperature (for a room or the whole house, depending on your heating system), while they use a temperature sensor to measure the temperature in their location and adjust the operation of the HVAC system to keep that temperature.
- **Programmable thermostats** allow users to not only specify the current temperature of the room, but put the heating system on a schedule. Users can specify the times when they want rooms to be at a particular temperature, and the thermostat will automatically turn on the HVAC system at the right time. (E.g., lowering the temperature at 9am when everyone's at school or work and turning it up just before 5pm in time for everyone to return home.)
- **Wifi-enabled thermostats** allow users to remotely connect to their thermostat (e.g., by using an app) and adjust the temperature or schedule from outside the room or house.
- **Smart thermostats** (e.g. Nest, Ecobee) come in two different flavors:
    - Thermostats equipped with **occupancy detection features** (usually motion detectors) adjust the temperature based on whether there are actually people in the room (e.g., overriding pre-set heating schedules if a room is actually unoccupied).
    - **Learning thermostats** draw on occupancy (and other) data collected about how a home's inhabitants actually use their home (i.e., when and how much time they tend to spend in which rooms at what times, as well

as their temperature preferences) and make a model to predict the most appropriate temperature for each part of the home at any given point in time.

What Makes a Thermostat Smart?

Precise Adjustments: Smart thermostats promise to help users save energy on heating and cooling their homes. This is based on the assumption that they make more precise and fine-grained decisions about when to turn the heating or cooling up or down compared to manual user control. For an overview of smart thermostat benefits, read this article.

Balancing the Grid: Smart thermostats often come with network interfaces that can be accessed remotely. Utility companies have started to harness this feature to defend against power outages caused by *expected* usage peaks which threaten to destabilize the electric grid. They incentivize customers (offering one-off premiums or reduced rates) to sign up for "demand-response programs" which allow utility companies to proactively balance electricity usage in anticipation of demand peaks by remotely adjusting AC or heat pump settings (raising or lowering the target temperature for the home) to reduce the load on the grid at times of high demand.

Data Analysis: Like many "smart" technologies, smart thermostats are smart because they collect, store and analyze data and adjust their settings based on the conclusions drawn from that analysis. In the case of smart thermostats, this data primarily concerns users' temperature-related preferences, as well as their habits and behavior in their home (e.g., when they get up, when they leave the house, which rooms or floors are in

use at which time of the day). In short, they gain insights which many of us would intuitively consider to be private (you can read reviews of different smart thermostats' "creepiness" on this website). Which privacy-invasive insights can be derived through a smart thermostat depend on the design of the thermostat, i.e., on decisions about what data is collected, at which intervals, how it is stored and processed and who will or might get access to it.

You might wonder: what can data from inside a home actually betray? How might knowing the movements of people within their homes (among other information) be used by those who have access to it? Read this abridged article about the tactics used by the East German secret policy (Stasi) to spy on and undermine opposition movements and other dissent. The Stasi routinely surveilled individuals' homes to gain information on their private lives which could be used to infer whether they were a part of dissident groups and activities, and to blackmail them or otherwise disrupt their lives to undermine those activities. We'll come back to this case-study about surveillance later in the assignment.

A Privacy-Preserving Smart Thermostat

In this assignment, you will explore this problem of privacy in greater depth. You will design and write a product specification for a smart thermostat which is supposed to be as protective of users' privacy as possible. The following tasks will lead you through a privacy threat-modeling process in which you will be able to practically apply the ideas and concepts we have discussed in class and reflect concretely on **what it would take to design a smart thermostat which puts privacy front and center.**

We have outlined the required features and hardware specification for your smart thermostat below. To help frame your thinking, we have additionally listed some common features that you can choose to use (or not) to help you implement the required features. **Make sure to pay attention to the memory limitation in particular (only 256 MB)!** From a privacy perspective it might be ideal if you could create a smart thermostat that housed all user data in memory and ran the analysis of that data locally, but this would require a massive amount of storage. Memory is expensive and training machine learning models requires a huge amount of computational power. This constraint is at the core of your task: ***if it's not feasible to store all user data and perform all computations on the device, what measures can we take to prioritize and preserve privacy? What privacy- and functionality-related trade-offs do we have to make?***

Software Features and Hardware Specifications

Required Features:

The thermostat must

- Be able to be connected to "demand response" programs (see above)
- “Learn” occupancy schedule and temperature preferences of occupants and automatically adjusts temperature to to save as much energy as possible without sacrificing comfort
- Allow users to manually override temperature settings

Hardware Specifications:

- 256 MB flash memory
- Temperature Sensor
- USB Port

Other common design features of smart thermostats:

Below is a list of common components and features of smart thermostats to help get you started. *You may include some of these in your design, but you don’t have to.* Some may be necessary or desirable for your specific design, some may be superfluous, but all of them come with their own specific privacy-related trade-offs which you will have to analyze in later steps. Keep this in mind when you choose which features to use. You may also choose to include other components and features not listed here, as long as you can explain your choices and any privacy-related concerns when prompted as part of the assignment.

Technical means to detect occupancy:

- Occupancy sensors (most commonly motion detectors) in each room or in specific places across the home
- Magnetic sensors on doors which are triggered when doors are opened
- Geofencing (i.e., setting a virtual perimeter around the home which triggers an event, such as turning the heat up when a device is located – e.g., based on GPS coordinates – within the perimeter)

Control interfaces:

- Interface on device
- Smartphone app
- Built-in voice control
- Smart speakers (Google Home, Amazon Echo, …?)

Possible features:

- Provide information for the user (e.g., aggregate statistics about energy consumption at various points in time)
- Provide nudges ("You will save $10 if you turn down the temperature by 2 degrees")
- Allow for manual *remote* adjustment of temperature by users

Assignment Roadmap

Your task for this assignment is split into three parts which walk you through the process of designing your privacy-preserving smart thermostat. First, you'll go through a **privacy threat modeling process** and think about what data is collected, what information that data might reveal about a user, and how users might be harmed if that data were accessed by others (you'll complete most of this in the in-person lab session). Second, you'll select a few of the threats you've identified and write a short **specification and draw an architecture diagram** for a smart thermostat which mitigates those specific threats. Third, you'll **reflect on your design in light of a case study** to see how your smart thermostat might match up against real-world scenarios.

You should compile your work on these tasks in a separate document (Word, Google, other) and insert any figures or diagrams you make along the way! You're welcome to make a copy of this answer template we provided or write your own as long as it's clear what parts of the assignment your materials refer to..

Note: We know there's *a lot* of instructions below. We recommend you read through the instructions in full, then go through the assignment step by step. A lot of the text and instruction that follows consists of extra detail or tips to help you if you're stuck or confused!

Task 1: Complete the Privacy Threat Modeling Process

Building on your work in the lab session: In the lab session, you worked on documenting and categorizing privacy threats arising from a smart thermostat using a privacy threat matrix. Please include the privacy threat matrix you worked on in the lab section (or an updated version if you have used it for further brainstorming since) in your submitted document. (We will not grade your matrix, but it will be helpful to understand your

reasoning in the following task.) Now draw on your work from the lab to complete the last step of the privacy threat modeling process:

Use your judgment to decide which privacy threats you intend to make a priority to mitigate in the design of your smart thermostat. **List all the privacy-related threats you have identified on your privacy threat matrix and explain how they could be eliminated or mitigated in your design or why it would be hard or impossible to eliminate them. Point out *three* threats that you'll aim to mitigate in your design.**

Use the following prompts as a guide for your analysis.

a. Refer to the prompts in the guidance document (see Guidance for step 4) to identify different approaches to mitigation (e.g., focusing on the amount of data collected vs. storage, or access control).
b. Identify any trade-offs (in terms of functionality or other characteristics) you have to make in order to mitigate a specific privacy-related threat.
c. Are there privacy threats you explicitly choose not to mitigate or which your design cannot mitigate? (That is fine and expected.) If so, provide a justification for your choice. (E.g., why did you prioritize mitigating other threats, rather than this one? What would it take to mitigate it? Why do you think mitigating this threat is not worth sacrificing a particular functionality?)

Task 2: Mini Specification

Let's put all of this together! You have analyzed the different privacy-related threats and sketched a strategy for how to eliminate or mitigate them. In this second step, you will outline the technical details of your device and how they will help you preserve user privacy in a mini specification document.

For all the following steps, focus on the three privacy threats from Task 1 which you plan to mitigate in your smart thermostat. Make sure as you outline the technical details of your thermostat that you design with your chosen privacy threats in mind! Your specification should take the following form:

1. *General overview of your smart thermostat*:

   This should detail the features of your smart thermostat and what assumptions you are making about potential users and their needs. It should also explain what spaces the thermostat expects to operate in and have control over.

2. *Functionality requirements:*

   These functionality requirements should define the action space of your smart thermostat. In other words, you will need to write a requirement for each action that your smart thermostat can take. **It may help to think of actions as responses to an input that may or may not produce an output.** This includes both manual and automatic (or "smart") actions - a manual action is in response to a manual input from a user (think pushing a button on the thermostat, adjusting a setting in the thermostat app, or issuing a voice command), while an automatic action is taken by the thermostat based on automated input from sensors, databases, or algorithms.

   The motivation behind this is that it helps you understand how exactly your thermostat works and how to implement it. This helps you address any potential issues or threats in more detail before writing any code.

   Well-written functionality requirements are **clear** and **concise**. Don't try to cram a bunch of actions into one requirement!

   Here are some examples of what functionality requirements might look like for a non-privacy preserving smart thermostat.

   Manual actions:

   > Example: If the temperature_up button is pushed, the thermostat must increase the temperature by 1 degree, up to the maximum temperature.
   >
   > Example: If the temperature_down button is pushed, the thermostat must decrease the temperature by 1 degree, down til the minimum temperature.
   >
   > Example: If the thermostat reads a temperature in the room lower than the temperature set by the user, the thermostat must increase the temperature until the set temperature has been reached.

   Automatic actions:

   > Example: The thermostat must begin heating the home half an hour before the user arrives home, using previous arrival times to predict the user's arrival.
   >
   > Example: The thermostat must turn the system off to save energy when users aren't home.

Example: The thermostat must prepare a temperature schedule based on temperatures set by the user at different times of day. If the user's preferences change, the thermostat must prepare a new schedule.

Example [collecting motion data]: If the motion sensor in room B is triggered, it will generate a time-stamped, location-specific entry in the database.

Example [qualifying sensor actions]: If the motion sensor in room B is triggered, the thermostat must raise the temperature to the determined optimum daytime temperature, except between 10pm and 6am, when it must not override the automated schedule.

Note: You can copy/paste the examples here to get started on defining your manual and automatic action spaces. It'd be wise to spend most of your energy defining the smart/automatic action space!

3. *Architecture diagram*

An architecture diagram is a useful tool for visualizing how all the parts of a system or piece of software work together. As we've established above, in order to offer smart features, your thermostat is likely part of a larger data processing system. To think about privacy preservation, it's useful to visualize and consider the connections between your thermostat and the rest of the system.

In an architecture diagram, you **map out and explain relationships between distinct components** of your system. The architecture diagram we're asking you to make is called a block diagram.

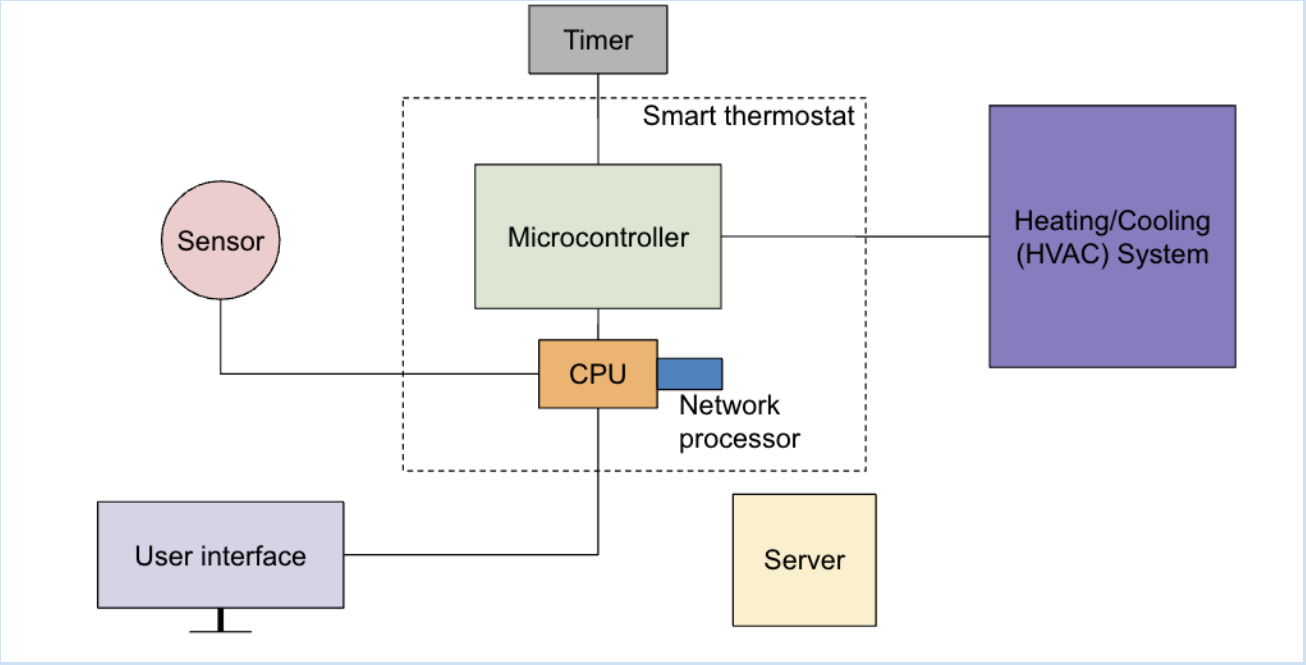
Architecture diagrams have different levels of granular detail, for yours follow these general guidelines:

1) If the components are not physically connected (except possibly by a wire), they are their own components on the architecture diagram
    a) Examples of distinct components: thermostat, remote motion sensors, and wireless microphones (think Google Home)
2) You can make assumptions about what kinds of infrastructure you have access to (data centers, large servers, etc.) and this will likely be the most abstract part of your diagram. That's expected and totally fine!

3) Use shapes/colors to represent different components and draw connections between them. Make sure to always label your components and label connections if useful or otherwise it would be unclear what the connection is.

Make a copy of the architecture diagram playground we've provided here and construct your architecture diagram!

Note: Does this all sound way too abstract? Here's a simple architecture diagram for a generic smart thermostat that should be useful!



4. *Privacy preservation overview*
    a. Now that you have some of the technical details outlined, review the three privacy threats you chose to address.
    b. Explain how and where your functionality requirements and architecture diagram effectively mitigate or address these privacy threats.
    c. **This is the place where you must provide additional privacy-related details on how the different components of your system operate and interact.** (E.g., details on duration and frequency of data collection and storage, general outlines of algorithms for setting appropriate temperatures, encryption, …)

Task 3: Case Study

You are almost done! In this final step, we want you to briefly assess how your privacy preserving smart thermostat design would perform in the context of a real-world case study.

Don't worry if your thermostat design reveals privacy vulnerabilities in the context of the case study. That is expected. The case study reflection is not a quality test of your design, but rather an opportunity to gauge and reflect on the limitations of privacy-conscious design.

Case study: Undermining political dissent in Eastern Germany

Earlier in the Lab 2 handout, you read an article on home surveillance by the East German Secret Police (Stasi). As a reminder, the article discusses how the Stasi used surveillance tactics to undermine their opposition and dissenting movements. They gained information on individuals' private lives by spying on their homes, then used that to infer whether they were a part of dissident groups and activities, and to blackmail them or otherwise disrupt those activities.

Briefly reflect on how vulnerable users of *your* smart thermostat design would be/would have been if the Stasi had been able to access the data the thermostat collects about users. Are there any design decisions you made which limit the amount of useful information that would be available to the Stasi in this case study? Are there residual vulnerabilities? (~ 5-6 sentences)

Handing in and Grading

Please type up your answers for all three tasks in a google or word document. Again, you're welcome to use the template we made if you find it useful. **Remember to insert your filled in privacy threat matrix and architecture diagram into your document!** When you're finished, submit your document on Gradescope.

Grading:

- Privacy threat modeling: **30%**
- Specification: **60%**

- Case Study: **10%**